# COMPUTATION OF PHYSICAL PROPERTIES OF MATERIALS USING PERCOLATION NETWORKS

WONG Yuk Chung

Supervised by
Professors WONG Chak Kuen and LEUNG Kwong Sak

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

©The Chinese University of Hong Kong
June 1999

The Chinese University of Hong Kong holds the copyright of this thesis. Any person(s) intending to use a part or whole of the materials in the thesis in a proposed publication must seek copyright release from the Dean of the Graduate School.

香港中文大學
圖書館系統
19 FEB 2000
UNIVERSITY
LIBRARY SYSTEM
THE CHINESE UNIVERSITY OF HONG KONG

**Abstract of thesis entitled:**

Computation of Physical Properties of Materials Using Percolation Networks

Submitted by Wong Yuk Chung

for the degree of Master of Philosophy

at the Chinese University of Hong Kong in June 1999

This thesis deals with the computation of physical properties of composite materials by simulated annealing based algorithms. The composite is represented by a bond percolation model. A physical model formed by two different kinds of materials with extreme physical properties is called a percolation network. We are interested in electrical and thermal properties of percolation networks. Firstly, we consider the simplest physical system that has one independent physical quantity. We apply a simulated annealing algorithm to calculate the electrical conductivity of percolation networks by the physical annealing schedule with an expected serial run-time $O(n \ln^2 n \ln(\ln n))$, where $n$ is the number nodes of a percolation network. Next, we extend our work on thermal properties. The work becomes more complicated because it involves three independent physical quantities to describe the physical system. In the computation process, we need not formulate the mutual inclusive relation on the global system in our approach. To calculate the thermal expansion coefficient, we design a parallel-simulated annealing algorithm to perform simulations. We implemented this algorithm on a 20-processor workstation. Comparing with a serial simulated annealing algorithm, the speed-up of the algorithm to simulate a network with 1024 nodes is about 7. Finally, we study the scaling behavior of thermoelastic moduli of percolation networks. To accelerate the computational speed for simulation, we propose and implement a conjectural parallel simulated annealing algorithm by the physical annealing schedules. Comparing with a serial simulated annealing algorithm, the speed-up of the algorithm to simulate a network with 1024 nodes is about 9 provided that a suitable conjectural method is used. After the simulations, we found that the thermal expansion coefficient of a percolation network has scaling behavior under certain conditions. Our simulation results have been validated and found to be very useful. We provide a universal algorithm to solve this type of problem.

# 摘要

本論文內容關於運用模擬退火爲本的演算法，去計算複合材料的物理特性。複合物是由一個鍵逾滲模型來描述。逾滲模型是指一個物理模型，由兩種擁有極端不同物理特性的材料所組成。我們感興趣探討的範圍，包括複合材料的電性和熱性性能。首先，我們考慮一個可以由單一物理量描述的簡單物理系統。我們運用適應式退火計劃的模擬退火法，計算逾滲網絡的導電性能。此演算法的期望運行時間爲 $O(n \ln^2 n \ln(\ln n))$ — $n$是逾滲網絡結的數量。在此基礎上，我們擴展研究範圍至熱性性能。由於涉及三種獨立不同的物理量來描述此物理系統，所以使這項研究愈具艱深。這方法的好處是在整個計算過程中，不需要考慮整個系統的相互關係。我們設計了一個平行模擬退火演算法，去計算熱膨脹系數。我們利用有二十個處理器的工作台，來執行此演算法。與串行模擬退火演算法作比較，在模擬一個擁有一千零二十四個結的網絡下，其執行演算法的速度提高了七倍。最後，我們對逾滲網絡的熱彈性模量之臨界現象，作深入研究。爲了提高模擬實驗的運行速度，我們建議及運用適應式退火計劃的初始值導向模擬退火法。與串行模擬退火演算法作比較，在有適當導向條件下，模擬一個擁有一千零二十四個結的網絡，其執行演算法的速度提高了九倍。透過模擬實驗，我們發現熱膨脹系數在某些情況下，有臨界現象發生。我們已驗証了模擬實驗的結果。綜合本篇論文，我們提供了一個通用的演算法，來處理此類問題。

# Acknowledgments

First of all, I would like to thank my supervisors Professors C. K. Wong and K. S. Leung for their guidance and patience. My research could not have been done reasonably without their insightful advice. I would also like to thank my internal markers Professors Y. L. Wu and K. H. Wong for their suggestion on my research project.

I want to express my thanks to our research group members S. K. Cheung, Y. B. Wong, and my fellow graduate students C. W. Fu, C. H. Cheng, K. W. Chu, T. S. Tam, Y. K. Yung, and W. W. Kwong. They opened up a new angle for me to view my work, thereby allowing me to make a more thorough treatment of the subject.

My great gratitude goes to my undergraduate supervisor Professor F. G. Shin who is full of enthusiasm in research. He enlightened me on some other physical properties of the work.

Finally, I would like to thank my parents and family. Without their support, I could not make all these happen.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The idea of percolation arises in considering a fluid contained by a partly porous wall. In such a wall truncated channels of various lengths and directions exist, and if a continuous connected path of channels linking the inner and outer surfaces occurs, fluid can pass through the wall — it percolates. Such behavior is of interest, for example, in hydrologic studies and in petroleum engineering. A specimen of material might be tested for its percolating property, critical porosity, by injecting mercury into it and searching for an electrical conducting path across it. A similar occurrence of electrical current flow will exist in an insulating medium that contains randomly distributed metallic inclusions when the volume fraction of the inclusions is high enough. The concept has also been exploited in studying phenomena as diverse as the percolation of charged particles, transported along the tangled magnetic field lines frozen into plasma and the configuration of galactic clusters and voids in the universe. We can see that the topics of percolation have wide-spread applications in different fields.

## 1.1 Motivation

Up to now, the accurate prediction of composite materials is very difficult outside the range of laboratory experience, especially on a composite governed by percolation behavior. Take a typical example, when an electronic composite is composed of insulator as a matrix and conductor as a filler, how could you predict the overall electrical behavior (effective electrical properties) of the electronic composite? It is very difficult to make a good prediction on the electrical properties of this type of electronic composite. The overall electrical behavior of the electronic composite is governed by percolation behavior through the formation of the randomly distributed conductive particles throughout the insulating matrix. To represent a system with percolation behavior, physicists and mathematicians establish various models to describe the properties of materials with such behavior.

For the computation of the special materials represented by physical models, a sophisticated algorithm that is fast as well as efficient in the use of memory is needed. Typically, the model consists of small nodes (sites) to form a system. At least thousands of nodes in a system are required to do a stochastic simulation. The time complexity of an algorithm to simulate the system depends on the complexity of the physical model, the number of independent physical properties involved in the problem and the dimensions of the model. Therefore, to study the percolation problems using percolation models, a robust, efficient, accurate algorithm is very essential. However, existing algorithms do not meet these requirements and some assumptions are needed on the physical models to simplify the complexity of computation.

## 1.2 The Scope of the Project

To solve the problem, we study the percolation behavior using percolation networks and propose an algorithm to compute the physical properties of percolation networks. Clearly, the objectives of our project are

- to aid the design and construction of composite materials with predetermined physical properties using percolation networks;
- to design an efficient stochastic algorithm to compute effective properties of composite materials;
- to predict the percolation effect of physical properties of composite materials;
- to demonstrate the flexibility of our proposed algorithm and model;
- to find an optimal annealing schedule for this kind of problem.

In this project, the physical properties of interest are electrical conductivity and thermal expansion coefficient. Both of them are important parameters in physics and material science. The other physical properties can be obtained by our algorithms with suitable modifications.

Also, we will focus on using a percolation model (percolation network). A real world system having percolation behavior will not be discussed in detail but an example will be given to demonstrate the relation between a real world system and a physical model in Appendix A.

## 1.3 An Outline of the Thesis

After we have introduced the work of this project, we outline the remainder of this thesis.

In Chapter 2, we review the background of percolation and simulated annealing. Both the origin of the problem and the recent research interest of the problem will be fully discussed. The basic concepts of percolation effect will be given in this chapter. There exist many physical models to describe percolation effect. The typical percolation models will be introduced with some simple examples. In our project, we propose a stochastic technique, a simulated annealing based algorithm, to solve the percolation problems that

are specified in later chapters. Therefore, the algorithms of simulated annealing will also be briefly reviewed. The challenge of this project can be seen in this chapter.

In Chapter 3, we introduce a common physical property - electrical conductivity that is computed using percolation networks. Electrical conductivity is an important parameter in physics and material science. The electrical property of a percolation network is computed by our proposed stochastic approach, where iterative process of solving systems of linear equations and highly non-linear time complexity methods are replaced by a simulated annealing method. The simulation results and the comparisons of traditional methods and our proposed method will be discussed.

In Chapter 4, we present a parallel stochastic technique to calculate the thermal expansion coefficient of a two-dimensional percolation network. The physical model and the simulation method will be given in this chapter. We implemented the parallel algorithm on a 20-processor workstation and the simulation result will be shown.

In Chapter 5, we propose a conjectural parallel simulated annealing algorithm to accelerate the speed of computation. Based on this algorithm, we study the critical properties of the thermoelastic moduli of percolation networks. The experimental result and the speedup of the algorithm will also be given.

# Chapter 2

# Related Work

In the field of physics [1] and material science [2], percolation is a critical phenomenon. The concept of percolation has been closely associated with the permeation of fluid porous media. This phenomenon can be described by the random agitation of particles as in Brownian motion and diffusion. Percolation thresholds were first studied formally by Broadbent and Hammersley [3], who introduced lattice models for the flow of a fluid through a static random medium, and showed rigorously that no fluid will flow if the concentration of active medium is smaller than some non-zero threshold value. This kind of problem is called percolation process.

## 2.1 Percolation Effect

A similar occurrence is found in electronic composites and it is detailed by [1]. Let us consider a composite that is composed of insulator as a matrix and conductor as a filler has the volume fraction of conductor $v$. At first, $v$ equals to zero. When a voltage is applied to the composite, no electrical current flows into the composite. When we are increasing the volume fraction $v$ at a certain value, the current reading between the composite will

suddenly increase. It is because there exists a 'conducting path' to let the current flow into and out of the composite. This phenomenon is called percolation effect.

There are various models to describe percolation effect. The models typically use the occupied probability of conductor $p$ instead of the volume fraction of conductor $v$. Actually, there are many different models to connect the relation between $v$ and $p$. One of models in [4] has been selected to demonstrate how to relate $v$ and $p$. Since this part is out of the scope of our project, we show the work in Appendix A.

This project deals with the peculiar phenomena of percolation near the concentration $p_c$ where for the first time a percolating cluster is formed. All sites within one cluster are thus connected to each other by one unbroken chain of nearest-neighbor links from one occupied bond to a neighbor bond also occupied. These aspects are called critical phenomena, and the theory to describe them is scaling theory. There are many different kinds of models [1, 5] in this field, such as bond percolation [6, 7, 8], site percolation [9], continuum percolation [10] and so on. Our studied percolation system will be described in Section 2.2.

## 2.2 Percolation Models

In this section, we are going to introduce two percolation models. The first one is a site percolation model; the other one is a bond percolation model.

### 2.2.1 Site Percolation

For the ease of demonstration, a simplified demonstration from [9] is done to explain why this is a challenge topic for research. Let $p$ be the probability that the site will be occupied and a number that is larger than zero is marked. The number 0 means the site has not occupied. If a cluster (with the same spanning non-zero number) spans over from the one

parallel side to another, it is said to be percolation. The value of $p$ will increase with each time step until the existence of a spanning cluster. In other words, no percolation is exist at $p < p_c$. A simple demonstration is shown in Figure 2.1. The five experimental results are listed in Table 2.1. It can be seen that the value of $p_c$ is very hard to estimate.

| | | | | | |
|---|---|---|---|---|---|
| 1 | 0 | 2 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 5 |
| 0 | 5 | 5 | 5 | 0 | 5 |
| 5 | 5 | 0 | 5 | 5 | 5 |
| 0 | 0 | 6 | 0 | 5 | 0 |
| 0 | 0 | 6 | 6 | 0 | 7 |

Figure 2.1: A two-dimensional $6 \times 6$ site percolation model.

After this simple experiment, it can be seen that the value of $p_c$ depends on the symmetry of the lattice on its dimension. In addition to the square lattice, the most common two-dimensional lattice is a triangular lattice. The essential difference between square and triangular lattices is in the number of nearest neighbors.

Table 2.1: The value of $p_c$ in 5 times simulations with $N = 20$, $N = 30$ respectively.

| $N = 20$ | $N = 30$ |
|---|---|
| $p_c$ | $p_c$ |
| 0.5040 | 0.5190 |
| 0.5230 | 0.4790 |
| 0.5390 | 0.4730 |
| 0.4450 | 0.4710 |
| 0.5230 | 0.5050 |

### 2.2.2 Bond Percolation

Bond percolation model is a common model to study percolation effect. If the model is formed by sites (nodes) with regular pattern, it is called a network. The network can describe percolation effect called a percolation network.

In this project, we are interested in electrical property and thermal properties. For the electrical property, a percolation network means a lattice constructed of a random mixture of conducting and non-conducting links. For the thermal properties, a percolation network means a lattice constructed of a random mixture of high thermal expansivity and low thermal expansivity links. We mainly study two physical quantities in our project. One is electrical conductivity; the other is thermal expansion coefficient.

Since our main objective is to search an efficient algorithm to solve this type of problem, we will only deal with two-dimensional percolation networks. There are many researchers dealing with the problems in two-dimensional space [6, 7, 8, 9, 11]. In the later chapter, you will find that the implementation of our algorithm will not depend on the dimension of model.

In this project, the selected lattice is in a square sharp. Although the selection of the lattice sharp will affect the value of $p_c$, our proposed algorithm is very flexible. The shape of a lattice will not affect the implementation of the algorithm.

## 2.3 Simulated Annealing

In our project, we propose a stochastic algorithm, which is simulated annealing based algorithm to compute the physical properties of percolation networks. Simulated annealing (SA) was introduced by Kirkpatrick *et al.* [12], and by Cerny [13] as an efficient stochastic algorithm for solving combinatorial optimization problems, such as the traveling salesman problem. Their development of the SA algorithms is based on the principle of Metropolis algorithm.

The term SA derives from the roughly analogous physical process of heating and then slowly cooling a substance to obtain a strong crystalline structure. In simulation, a minima of the energy function (cost function) corresponds to this ground state of the substance. The SA process lowers the temperature by slow stages until the system "freezes" and no further changes occur. At each temperature, the simulation must proceed long enough for the system to reach a steady state or equilibrium. This is known as thermalization. Specifically, the system must fulfill the criterion of detailed balance [14, 15]. The time required for thermalization is the decorrelation time; correlated microstates are eliminated. The sequence of temperatures and the number of iterations applied to thermalize the system at each temperature comprise an annealing schedule.

To apply SA, the system is initialized with a particular configuration. A new configuration is constructed by imposing a random displacement. If the energy of this new state is lower than that of the previous one, the change is accepted unconditionally and the system is updated. If the energy is greater, the new configuration is accepted probabilistically. This is the Metropolis step, the fundamental procedure of SA. This procedure allows the system to move consistently towards lower energy states, yet still escape from local minima due to the probabilistic acceptance of some upward moves. If a suitable annealing schedule is selected, SA guarantees an optimal solution with certain probability [14]. A typical SA algorithm is shown in Figure 2.2.

Also, one of the most important implementation parameters of an SA algorithm is the neighborhood structure. In our project, the magnitude order of the number of the neighborhood is small. Each node of the percolation network has four nearest neighbors only because the network is a square lattice structure. Even a triangular lattice structure, the network has six nearest neighbors. Therefore, this type of problem is suitable for SA algorithm.

```
Procedure SA algorithm
Begin
   t <- to;                (* t annealing temperature *)
   P <- Po;                (* P configuration *)
   E <- Cost(P);           (* Cost() cost function *)
   while not stopping-criteria()
      P' <- Generating(P);
      E' <- Cost(P');
      Delta <- E' - E;
      if (Delta <= 0) or (random() <= exp(-Delta / t))
         P <- P';
         E <- E';
      t <- reduce-temperature(t);
   End while;
End
```

Figure 2.2: A typical simulated annealing algorithm.

# Chapter 3

# Electrical Property

Electronic composites have been played an important role in electronic packaging and devices, such as sensors, resistors, and transducers [16, 17]. When an electronic composite is composed of insulator as a matrix and conductor as a filler, the overall electrical behavior (effective electrical properties) of the electronic composite is governed by *percolation behavior* through the formation of the random distributed conductive particles throughout the insulating matrix. Therefore, it is very difficult to make a good prediction on the electrical properties of this type of electronic composites. A physical system formed by two different kinds of materials with extreme physical properties is called a percolation network. For the electrical properties, a percolation network means a lattice constructed of a random mixture of conducting and non-conducting links.

## 3.1 Electrical Conductivity

Conductivity is one of the common physical quantities in electricity [18]. The ratio of the electric current density $J$ to the electric field $E$ in a material is called electrical conductivity

$g$ or simply called conductivity in the rest of this paper. The constitutive equation is

$$J = gE. \tag{3.1}$$

The reciprocal of the conductivity is called the resistivity ρ. This physical quantity is more familiar to people.

Consider a finite lattice for an $L \times L$ resistor network, where $L$ is the number of the site, be constructed of bonds chosen randomly to have conductance $a$ with probability $p$ or $b$ with probability $q$, where $p+q=1$. Without loss of generality, we choose $a < b$. In the limit that $L$ is very large, the specific conductivity of such a system comes to have the character that the sample-to-sample variation becomes vanishingly small, and we may refer to the specific conductivity of a given specimen as if it depends only $a$, $b$ and $p$. This explains why the value of $L$ should be larger as possible. In the case of $a << b$, and $b$ finite, there is a threshold concentration $p_c$; such that $p < p_c$, the lattice is so fragmented that it cannot conduct, and for $p > p_c$, there are connected conducting paths of larger extent. Furthermore, it has been noticed that the conductivity or electrical physical quantities obeys a power-law relation $(p-p_c)^t$ for $p > p_c$ in experiment [2], where $t$ is a parameter.

In this project, one of the physical properties of interest is electrical conductivity. The other type of conductivity, say thermal conductivity, can also be calculated through the same method. Mathematically, effective medium analyses of the electrical resistivity (inverse of conductivity) is equivalent to thermal conductivity, dielectric constant (or permittivity), and magnetic permeability of a composite. It can be traced back to the classical works of Rayleigh [19] and Maxwell [18].

After established the relation between the volume fraction $v$ of the filler material and the probability $p$ of bonds occupying the site in the percolation networks, we can determinate the conductivity of the composite with a given volume fraction of filler. Actually, there are many different models to connect the relation between $v$ and $p$ in the percolation networks. One of models [4] has been selected to demonstrate how to relate $v$ and $p$. Since this part

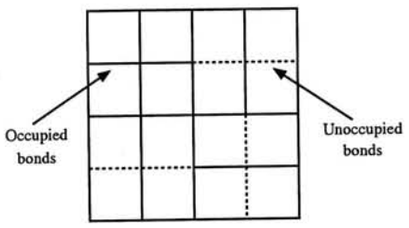



Figure 3.1: A two-dimensional random resistor network used in the computation with $L = 5$.

is out of the scope of our project, we show the work in Appendix A.

## 3.2 Physical Model

An important critical exponent for percolation is the conductivity exponent $t$ defined by

$$\sigma \sim (p - p_c)^t, \tag{3.2}$$

where $\sigma$ is the conductance (or inverse resistance) per unit length in two dimensions. There are various models of this type of problem [7, 9]. They consider a dual square lattice and use the Laplace's equation to solving the conductivity of the resistor network. For simplicity, we consider bond percolation on a square lattice (Figure 3.1) where each occupied bond between two neighboring sites is a resistor of unit resistance $R$. Unoccupied bonds have infinite resistance.

Given a value of the bond percolation threshold $p_c$ on a square lattice, the critical exponent $t$ can be estimated in Equation (3.2) by averaging over many times spanning configurations for $p + \delta p$, where $\delta p$ is the very small value relative to $p$. A typical example that the conductance of a square lattice against $p$ is plotted in Figure 3.2. In two dimensions, $t$ is the same for lattice and continuum percolation. However, $t$ can be different in the case of three dimensions. This explains why there is much work [20] about the percolation networks in three dimensions.

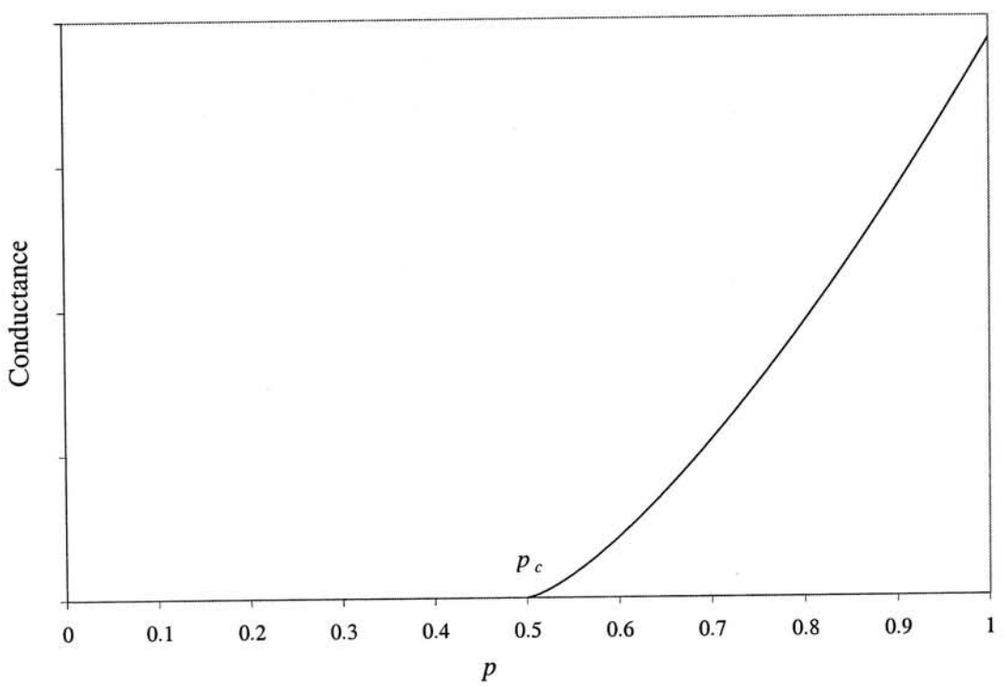



Figure 3.2: A graph of conductance $\sigma$ of a square lattice with $p_c = 0.5$ against $p$ and $t = 1.38$.

First principle calculation on the conductivity of percolation network bases on the Kirchhoff's voltage and current law. The potential of each node and the current passing through each resistor are then formulated a set of system equations. After solving those equations, the conductivity of a random resistor network can be calculated through Ohm's law.

However, this method is not efficiency. A set of equations needs to be formulated again ever the configuration of a random resistor is changed. It is not realistic to apply on a large system, say a network with $32 \times 32$ sites.

By Kirchhoff's law, the total current into any node must equal zero. Hence, the voltage at any site (node) is equal to the average of the voltages of all nearest neighbor sites connected by resistors (occupied bonds). By a relaxation method, the voltage at each site can be computed. Consider two-dimensional boundary value problems for $V(x,y)$, where $V(x,y)$ is the potential at the $x$-th column and $y$-th row node. We use a finite difference method and divide the space into discrete grid points. Let the grid spacing be $\Delta x$ in $x$-direction and $\Delta y$ in $y$-direction; $\pi$ be a bijective function. The potential $V(x,y)$ is expressed as

$$\begin{aligned} V(x,y) \approx & \frac{1}{N_e}\{\pi[V(x+\Delta x,y),i]+\pi[V(x-\Delta x,y),i] \\ & +\pi[V(x,y+\Delta y),i]+\pi[V(x,y-\Delta y),i]\}, \end{aligned} \tag{3.3}$$

where $N_e$ is the total number of all nearest neighbor sites connected by resistors. The term $\pi[V(x,y),i]$ means

$$\pi[V(x,y),i] = \begin{cases} i=1 & \pi = V(x,y), \\ i=0 & \pi = 0, \end{cases} \tag{3.4}$$

where $i$ is $0,1$. If $i=1$, it means the site connects with the corresponding nearest neighbor. If $i=0$, it means the site does not connect with the corresponding nearest neighbor. From Equation (3.4), $V(x,y)$ is the average of the potential at all connected nearest points. To

compute the conductivity for a given $L \times L$ resistor network (Figure 3.3), the voltage $V = 0$ is fixed at sites for which $x = 0$ and $V = 1$ is fixed at sites for $x = L+1$. In the $y$-direction, periodic boundary conditions are used. The voltage at all sites is then computed by using the relaxation method. The current through each resistor connected to a site at $x = 0$ is simply

$$I = \frac{\Delta V}{R} = \frac{V-0}{1} = V. \tag{3.5}$$

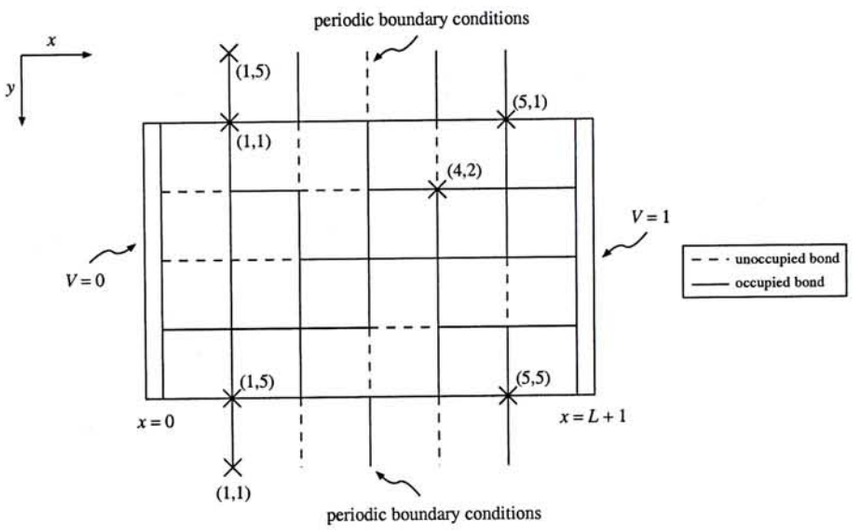



Figure 3.3: A two-dimensional random resistor network under applied voltage.

The conductivity is the sum of the currents through all the resistors connected to $x = 0$ divided by $L$. The algorithm of the relaxation method for computing the conductivity of a random resistor network on a square lattice is shown in Figure 3.4.

## 3.3 Algorithm

Other than that there are numerous methods to calculate the conductivity of a random resistor network. Another easier way of computing the conductivity, but indirect method, can be through the Einstein relation [21]. As we explain in Section 3.2, it can seen that $L$ should

```
PROCEDURE RELAXATION METHOD
Begin
   Initialize all the voltage of each node is zero
   Set the voltage V = 1 of each node at x = L + 1
   Set the voltage V = 0 of each node at x = 0
   For x in the range of the network
   For y in the range of the network
   Begin
   If Random < p then
      The bond is occupied
   Else
      The bond is unoccupied
   End
   Repeat
      Calculate the voltage of each node
   Until (no further change on the voltage) or
         (less than a pre-defined error value)
END
```

Figure 3.4: The algorithm of the relaxation method.

be larger as possible. But, a problem naturally comes with the increase of $L$. That is, the CPU time for computing the conductivity of a network also rapidly increases if a deterministic method, say relaxation methods, is applied in solving this problem. All these kinds of deterministic methods cannot be applied on a large system because of the non-linear time complexity on the number of sites. It is worth to develop an algorithm to calculate such kind of problems. Owing to this reason, we formulate this problem as an optimization problem and solve it by simulated annealing. In our problem, the conductivity of a random resistor network is computed from the process of a non-steady state to a steady state by an applied voltage at the parallel sides of the random resistor network with a suitable cooling schedule.

### 3.3.1 Simulated Annealing

Simulated annealing (SA) was introduced by Kirkpatrick *et al.* [12], and by Cerny [13] as an efficient stochastic algorithm for solving combinatorial optimization problems, such as the traveling salesman problem. Their development of the SA algorithms is based on the principle of Metropolis algorithm.

We now describe the simulated annealing algorithm that applies on this model. We define the objective function and neighborhood relation and minimize the cost of our defined objective function. Let us consider the configuration of conducting and non-conducting bonds as shown in Figure 3.3. At the beginning, the voltages at each node of the percolation network is zero. Then, the left side of the network is applied on 0 V and the right side of the network is applied on 1 V. We assume that the value of voltage on each node is discrete. The value of voltage is a sub-division by a grid of step size $w$. It is acceptable in a real world situation if $w$ is small enough.

Next, we apply a relatively higher potential $\Delta V$ on each node by choosing independently and randomly several Monte Carlo steps. This process is called symmetry breaking. This process is to ensure that the calculated value converges to our expected results. The adjustment of the voltage $\Delta V$ needs to follow three natural restrictions. If the adjustment of

the voltage violates the restrictions, no action will be taken on the selected node. The three restrictions are

1. The addition of the original voltage on a selected node $V$ and the adjustment of the voltage $\Delta V$ must be less than that of the applied voltage. The applied voltage in our case is 1 V.

2. The addition of the original voltage on a selected node $V$ and the adjustment of the voltage $\Delta V$ must be larger than that of a pre-defined minimum value. The pre-defined minimum value in our case is 0.00005 V.

3. The sum current of the percolation network must be less than a pre-defined bounded value. This value depends on the number of sites.

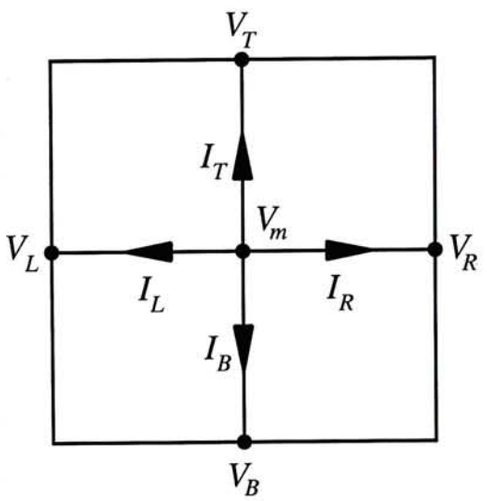



Figure 3.5: A node and its neighbors of the percolation networks.

### 3.3.2 Neighborhood Relation and Objective Function

Now, we are going to introduce the objective function and the neighborhood relation. Consider a node as shown in Figure 3.5. If the system is not in a steady state, there exists the

current difference $\Delta I$ at the node.

$$\begin{aligned} \Delta I &= I_T + I_B + I_L + I_R \\ &= V_T + V_B + V_L + V_T - 4V_m. \end{aligned} \tag{3.6}$$

Since we calculate the sum of total current difference $I$ at each node at the beginning, the sum of current difference at each node after one successful voltage adjustment step is $I + \Delta I - 4V_m + 4V_m'$, where $V_m'$ is the voltage value at the node after one successful voltage adjustment step. Since our objective is to reduce $\Delta I$ inside the percolation network due to the change of potential, we introduce the following objective function $\mathcal{E}$:

$$\mathcal{E} = \frac{1}{n} \sum_{i=1}^{n} |\Delta I_i|, \tag{3.7}$$

where $n$ is the total number of nodes and $i$ represents the corresponding node. As in the process of symmetry breaking, we apply a potential step $w$ on each node by chosen independently and randomly.

The algorithm is described more clearly in a simple mathematical expression. We want to minimize the objective function. Let the voltage value $V_m$ on a node at the $N$-th time step and $V_m'$ be the next voltage value at the $(N+1)$-th time step. Consider a single step of transitions; the probability $G[V_m, V_m']$ of moving from $V_m$ to $V_m'$ is given by

$$G[V_m, V_m'] = \begin{cases} \frac{1}{2N} & \text{if } V_m' \in \mathcal{N}_p, \\ 0 & \text{otherwise,} \end{cases} \tag{3.8}$$

where $\mathcal{N}_p$ is the collection of allowed configurations. The probability $A[V_m, V_m']$ of accepting $P'$ is

$$A[V_m, V_m'] = \begin{cases} 1 & \text{if } \mathcal{E}(V_m') - \mathcal{E}(V_m) \leq 0, \\ \exp[(\mathcal{E}(V_m) - \mathcal{E}(V_m'))/t] & \text{otherwise,} \end{cases} \tag{3.9}$$

where $t$ is a parameter, called temperature, that controls the acceptance-rejection probability. Therefore, $V_m'$ is accepted even $\mathcal{E}(V_m) < \mathcal{E}(V_m')$ if

$$\exp[\mathcal{E}(V_m) < \mathcal{E}(V_m'))/t] \geq \eta, \tag{3.10}$$

where $\eta \in [0,1)$ is a uniformly distributed random number. The probability Pr that the configuration $V_m$ transits to $V_m'$ is

$$\Pr[V_m \to V_m'] = \begin{cases} G[V_m, V_m']A[V_m, V_m'] & \text{if } V_m' \neq V_m, \\ 1 - \sum_{Q \neq P} G[V_m, Q]A[V_m, Q] & \text{if } V_m' = V_m. \end{cases} \tag{3.11}$$

Equation (3.11) means that the searching direction in space $S$ may not be in the direction of gradient descent. In other words, we can get the lowest cost value of $\mathcal{E}$ without being trapped by local minima if the adjustment of the parameter $t$ (cooling rhythm) is appropriate. The probability of ending in a global minimum is approaching to one.

### 3.3.3 Configuration Space

We define the initial configuration $V_{\text{init}}$ at this stage. Let the set of configurations that can be reached from $V_{\text{init}}$ by a sequence of moves of the nodes without any violation of the law of physics be $C_n$. The size of the configuration space $|C_n|$ is roughly bounded by

$$|C_n| < K \times (K-1) \times \ldots \times [K-(n-1)], \tag{3.12}$$

and where $K$ is the total number of grid points in the possible region that the nodes can move. Our objective is to find a configuration, such that the configuration has minimum net current value among the entire possible configurations. We assume

$$n = \frac{K}{c}, \tag{3.13}$$

where $c > 1$ is a constant. By Equation (3.12) and (3.13), the upper bound for $|C_n|$ is obtained

$$|C_n| < K^n \leq \exp[O(n \ln n)]. \tag{3.14}$$

### 3.3.4 Annealing Schedule

In this problem, we have to find a minimum value of $\mathcal{E}$. The value of $t$ should be decreased as the increase of time. Therefore, the rejection probability increases. There are many different kinds of annealing schedules [12, 14]. One of them is exponential decreasing schedule. The algorithm has one of control parameter $\gamma$ to control the decreasing rate of $t$.

$$\gamma = -\frac{1}{N_f} \ln(\frac{t_f}{t_0}), \tag{3.15}$$

and $t$ is expressed as

$$t = t_0 \exp(-\gamma N), \tag{3.16}$$

where $N_f$ is total number of iterations and $N$ is the $N$-th iteration and $t_0$ is the initial value of $t$ and $t_f$ is the final value of $t$. However, these methods may need the value $N_f$ and fix the cooling rate. The annealing temperature is reduced even the system does not reach in equilibrium. Therefore, the quality of the solution will be affected.

Ideally, we would like a method for determining the annealing schedule based on the behavior of the specific system. We would like to minimize the variation from equilibrium. During the annealing, we can measure the objective function and variance $C_H$ of the objective function at each value of $t$. These values can be used to determine the annealing schedule. However, since we are constantly changing $t$, it is not easy to obtain equilibrium values of $\mathcal{E}$ and $C_H$ for each value of $t$. But, we can estimate the values.

In our project, we are concerned with the solution quality rather than the run-time of the simulation, so we apply the physical annealing schedule. We denote $t_k$ as the range of

temperatures and $N_k$ as the number of Metropolis iterations at each $t_k$. We would like to choose $N_k$, so that thermal equilibrium is reached at $t_k$ before moving to $t_{k+1}$. We assume $\mathcal{E} = \exp(\mathcal{L}_k)$ and $\tau_{\exp}$ is the value of intersection point between the $x$-axis of $\mathcal{L}_k$ and the tangent line of $\exp(\mathcal{L}_k)$ at $\mathcal{L}_k = 0$, where $\mathcal{L}_k \in [0, N_k - 1]$. Then, we can estimate $\tau_{\exp}$ from measurements of $\mathcal{E}$ at smaller values of $\mathcal{L}$, and set $N_k \gg \tau_{\exp}$. Note that we should not set $N_k \geq 4\tau_{\exp}$ because fluctuations of $\mathcal{E}$ exist.

$$C_{H_k} = \frac{d\mathcal{E}}{dt} \approx \frac{\Delta\mathcal{E}_k}{\Delta t_k}, \tag{3.17}$$

$$t_{k+1} \approx t_k - \frac{\Delta\mathcal{E}_{k+1}}{C_{H_k}}. \tag{3.18}$$

By a suitable initial value of $\Delta t_0$ at $k = 0$, we can minimize the objective function through this simulated annealing algorithm. The final temperature of SA algorithm is determined by fixing

- the number of temperature values to be used, or
- the total number of solutions to be generated.

### 3.3.5 Expected Time Bound

In [11], it points out that the polynomial time bounds which could be derived from the theory of rapidly mixed Markov chains [22, 23] are still relatively large and it provides an alternative method on the calculation of the time bound. In this paper, we apply the similar technique in [11] to find the time bound.

We define the starting value of temperature $t(0)$, i.e., the initial annealing temperature. The length of Markov chains $L_k$ is for a fixed temperature value at the $k$-th annealing step. The simulation will stop at the final temperature $t(k_f)$. At first, the value of $t(0)$ has to let

all transitions from $V_m$ to $V'_m$ [14]. Hence,

$$\begin{aligned} \exp(-\Delta\mathcal{E}_{\max}/t(0)) &\approx 1, \\ \exp(-\Delta\mathcal{E}_{\max}/t(0)) &= c_1, \\ t(0) &= -\frac{\Delta\mathcal{E}_{\max}}{\ln(c_1)}, \end{aligned} \tag{3.19}$$

where $c_1$ is a positive value that is close to 1. $\Delta\mathcal{E}_{\max}$ is the possible value of the maximum difference of the objective function.

Now, we define the two values considered for the actual length of Markov chains $L_k$.

$$L_k{}^{(1)} = l(2n), \tag{3.20}$$

$$L_k{}^{(2)} = \exp(\Delta E_{\max}/t(k)), \tag{3.21}$$

where $l$ is a rational number and $l < 1$ for large $n$. According to [11], if the expected number of trials necessary to leave a given configuration is larger than the actual value $L$ for the length of Markov chains, it is indeed the time to finish the procedure of simulated annealing. Hence, the stopping criterion is obtained from Equation (3.20)

$$t(k_f) \leq \frac{\Delta\mathcal{E}_{\max}}{\ln 2nl}. \tag{3.22}$$

By a suitable value of $c_2$, we can express Equation (3.18) into the following relation:

$$(c_2)^{k_f} \cdot t(0) \leq t(k_f), \tag{3.23}$$

$$(c_2)^{k_f} \cdot \frac{-\Delta\mathcal{E}_{\max}}{\ln c_1} \leq \frac{\Delta\mathcal{E}_{\max}}{\ln 2nl}, \tag{3.24}$$

where $c_2$ is a positive constant and close to one. Finally, we obtain

$$k_f \leq \frac{1}{\ln c_2} \cdot \ln(-\frac{\ln c_1}{\ln 2nl}). \tag{3.25}$$

From Equation (3.25), we can see that the number of cooling steps does not depend on the objective function. If the length of Markov chains is $L$, the algorithm undergoes $L \cdot k_f$ successful moves.

Let $\mathcal{R}_k$ denote the expected ratio of the entire number of processed trials to the number of accepted moves $N_k$ at the $k$-th annealing step with annealing temperature $t_k$. The average value of the expected ratio $\overline{\mathcal{R}}$ is given by $\frac{1}{k_f}\sum_k \mathcal{R}_k$. For a constant length $L$ of Markov chains, the total number of processed trials from $t(0)$ to $t(k_f)$ is given by $\overline{\mathcal{R}} \cdot L \cdot k_f$. In each trial, the running time of $\mathcal{T}$ is needed to perform the computation which includes

- the random selection of a node,
- the calculation of local voltages,
- the calculation of the difference of the objective function,
- the updating of the objective function if necessary, and
- the updating of annealing parameters if necessary.

In the calculation of the local stresses, a node is selected at random with a direct access to the address containing the node's information. If we assume that coordinates are represented by $\lg x$ and $\lg y$ bits and a square complexity for the basic arithmetic operations in this computation, $\mathcal{T} = O(\ln^2 K)$. Taking logarithm on both sides of Equation (3.14), we get $\mathcal{T} = O(\ln^2 n)$. If we assume $L = O(n)$, the expected running time $\mathcal{T}_e$ is

$$\begin{aligned} \mathcal{T}_e &\leq L \cdot k_f \cdot \overline{\mathcal{R}} \cdot \mathcal{T}, \\ &\leq \frac{2nl}{\ln c_2} \cdot \ln(-\frac{\ln c_1}{\ln 2nl}) \cdot \overline{\mathcal{R}} \cdot \mathcal{T}, \\ \mathcal{T}_e &= O(n \ln^2 n \ln(\ln n)). \end{aligned} \tag{3.26}$$

## 3.4 Results

In this section, we will show the simulation results of a percolation network with $32 \times 32$ sites because it is large enough to show the advantage of this algorithm but small enough to compute the result within a reasonable limited time. We have implemented the algorithm on a Sun SPARCstation 20. The run-times on calculating the corresponding physical properties for different numbers of nodes are shown in Table 3.1. In Table 3.2, we select the simulation results on computing the conductivity $\sigma$ of percolation network with 900 nodes. The bonds are chosen randomly to have finite resistance ($1\,\Omega$) with probability $p$ and to have another resistance ($\infty\,\Omega$) with probability $q$. We can validate these results by noting that they are very close to those shown in [24]. The value of resistance is chosen as $1\,\Omega$ and the applied voltage is 1 V. It can be obtained the simplified relation or equation, e.g., Equation (3.5).

In Figure 3.6, it shows the conductivity in a $32 \times 32$ resistor network against bonds occupied probability $p$. The line is the trend of the data and the data is fitted by a power law. The equation is shown in Figure 3.6. The result is consistent with [2] by applying a suitable relation between $v$ and $p$.

In Figure 3.7, it shows the voltage distribution on a percolation network with $32 \times 32$. In the $x$-axis direction, the value $x = 1$ in Figure 3.7 is the value $x = 0$ in Figure 3.5. In the $y$-axis direction, the value $y = \mathrm{S1}$ in Figure 3.7 is the value $y = 1$ in Figure 3.5. The rest of the corresponding positions can be found by the similar transformation. From this figure, it can be seen that the voltage values do not evenly distribute over the network. This explains why the conductivity is so hard to be obtained without heavy CPU time. Therefore, a fast algorithm, such as this algorithm, is very needed to solve a percolation network problem.

The time complexity of relaxation method is $\Omega(L^2N)$, where $L^2$ is the number of sites and $N$ is number of iterations. From our experiment, the rate of growth of $N$ is $\Omega(L^2)$. The total time complexity of relaxation is $\Omega(L^4)$. The expected run-time of simulated annealing is $O(n \ln^2 n \ln(\ln n))$. Compared with the over-relaxation method, the CPU time on computing the conductivity of a large system is reduced. According the results, in the

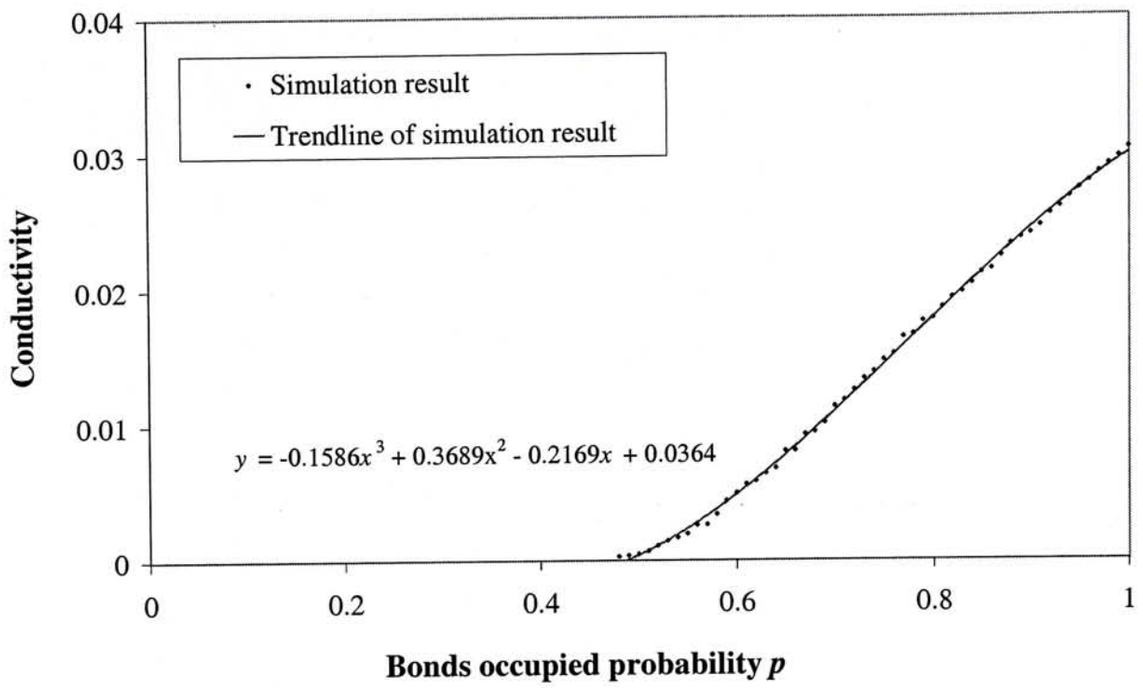



Figure 3.6: A graph of conductivity in a $32 \times 32$ resistor network against bonds occupied probability $p$.

case of $32 \times 32$, the CPU time reduction is about 10 to 16 percent.

Table 3.1: A comparison between the network with 400 nodes and with 900 nodes.

| Number of nodes | 400 | 900 |
|---|---|---|
| Approximate run-time / min (conductivity) | 5 | 9 |

## 3.5 Discussion

We have performed stochastic simulations of a two-dimensional percolation network with mixed conducing and non-conducing bonds, based on simulated annealing by a physical annealing schedule. We want to point out that this algorithm can be used to solve a general

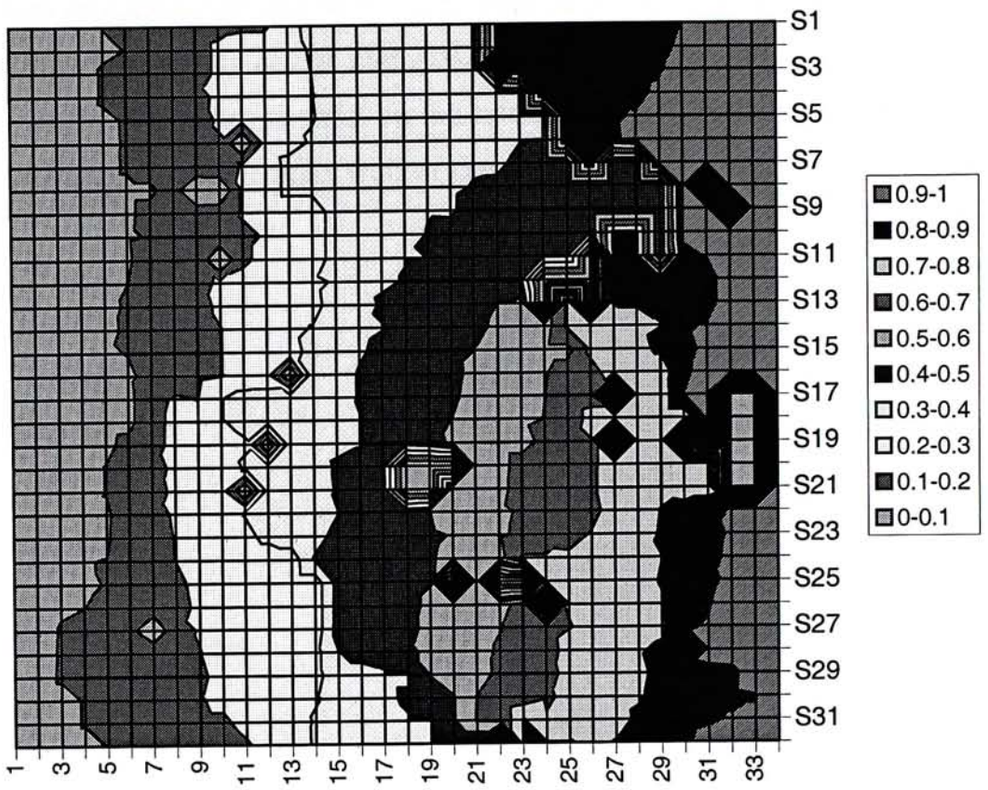



Figure 3.7: Voltage distribution on a percolation network with $32 \times 32$ sites

problem that cannot be done by relaxation method if the non-conducing has a finite resistance value, say $10000\,\Omega$. But, our algorithm can find the conductivity by changing the neighboring site definition. In this project, we select a square percolation network model to calculate its conductivity. Actually, there are other common lattice shapes of percolation networks, say honeycomb or triangular. But, in these shapes, the ease of formulation of the relaxation method will lose. But, it does not affect in our algorithm. It is because we only have to re-define a neighborhood structure. Importantly, our algorithm has the potential

Table 3.2: Simulation results on the conductivity of a percolation network.

| $p$ | 0.51 | 0.54 | 0.57 | 0.6 | 0.63 |
|---|---|---|---|---|---|
| $\sigma$ / $\Omega^{-1}m^{-1}$ | 0.000682 | 0.001692 | 0.002622 | 0.004966 | 0.006418 |

to handle a percolation network with more than two different types of bonds. This problem may not be handled by others methods. From the study, we can see that our proposed simulation method does not depend on the lattice shape. We need not to re-formulate the equations of a physical system, as well as the re-write a new computer program.

# Chapter 4

# Thermal Properties

In the previous chapter, we have calculated the electrical properties of percolation networks by a simulated annealing algorithm. As composite materials have been played an important role in our everyday lives, there are many useful physical properties like electrical conductivity.

Thermal expansion coefficient is also an important parameter in composite materials [25, 26]. In this chapter, we propose a stochastic approach to calculate the thermal expansion coefficient of a percolation network. The problem will then become more complicated. It is because we need three independent parameters to describe the physical properties of this kind of system.

At first, the definition of the thermal expansion coefficient of material will be given. Also, the terminology of our problem will be introduced, including the definition of thermal expansion coefficient, and the corresponding physical quantities. Next, we will introduce our physical model that uses in our project. Then, we will describe our problem and the implementation of this model by simulated annealing. Also, the parallelization technique will be discussed. We will show the implementation and our experimental results in Section 4.4. Finally, the importance of this algorithm and the result will be discussed.

After reading this chapter, you will get the idea that percolation is a problem, which is, in principle, easy to define. However, it is not so easy to solve.

## 4.1 Thermal Expansivity

There are many papers [25, 26] to study the thermal expansion coefficient of composite materials. A change in size of a body at constant pressure due to varying temperature is referred to as thermal expansion [27]. In other words, we want to know the difference of the length deformation of an object due to the change of temperature. If we give you a rod of negligible thickness and the rod is formed by pure material, you can answer the question without any problem. But, how do you know the deformation of a rod that is formed by composite materials outside the range of laboratory experience?

Before, we answer the question. We define the physical quantities first. Let us consider deformations that are accompanied by a change in the temperature of the body. This can occur either as a result of the deformation process itself, or from external causes. We shall assume the body is in an un-deformed state in the absence of external forces at some given temperature $T_0$. If the body is at a temperature $T$ different from $T_0$, then, even if there are no external forces, it will in general be deformed, on account of thermal expansion. In the expansion of the free energy $F(T)$, there will therefore be terms linear, as well as quadratic, in the strain tensor. Owing to the limitation of the components of the tensor, we can form only one linear scalar quantity. We shall also assume that the temperature change $T - T_0$ which accompanies the deformations is small. We can then suppose that the coefficient of $\varepsilon$ in the expansion of $F$ (which must vanish for $T = T_0$) is simply proportional to the difference $T - T_0$. In free thermal expansion of the body (external forces being absent), there can be no internal stress. Then, we get

$$\varepsilon = \alpha(T - T_0). \tag{4.1}$$

$\varepsilon$ is the relative change in length caused by the deformation. Thus $\alpha$ is just the *thermal expansion coefficient* of the body.

For the elastic properties [11], a percolation network means a lattice constructed of a random mixture of rigid and non-rigid links. For the thermal expansion properties, a percolation network means a lattice constructed of a random mixture of relatively high thermal-expansion-coefficient and relatively low thermal-expansion-coefficient links.

## 4.2 Physical Model

In the literatures [28, 29, 30], a percolation network with the underlying hexagonal-like network structure was studied to calculate its elastic properties. In [31, 32], not only were the bond-stretching forces calculated, but the angle-bending forces were also calculated in the network. But, for simplicity, we do not take into account the angle-bending forces in our model.

### 4.2.1 The Physical Properties

Consider a finite lattice for an $m_r \times m_c$ percolation network and let the total number of sites be $n$. The network is constructed of bonds chosen randomly to have thermal expansion coefficient $\alpha_1$ with probability $p$ or $\alpha_2$ with probability $q$, where $p+q=1$. Without loss of generality, we choose $\alpha_1 > \alpha_2$. For the ease of implementation, we form a square lattice within a square boundary. Hence, $m_r$ and $m_c$ have the same value $\sqrt{n}$, where $\sqrt{n}$ are positive integers. In Figure 4.1, it shows the square configurations of the $\alpha_1$ and $\alpha_2$ bonds with $20 \times 20$ sites. The node is the conjunction of the bonds. The black solid line represents the $\alpha_1$ bond and the partially gray filled line represents the $\alpha_2$ bond.

Consider an infinitesimal rectangular element as shown in Figure 4.2. Let $\Delta u$ and $\Delta v$ be the relative infinitesimal displacement along the $x$-axis and $y$-axis respectively. The strain

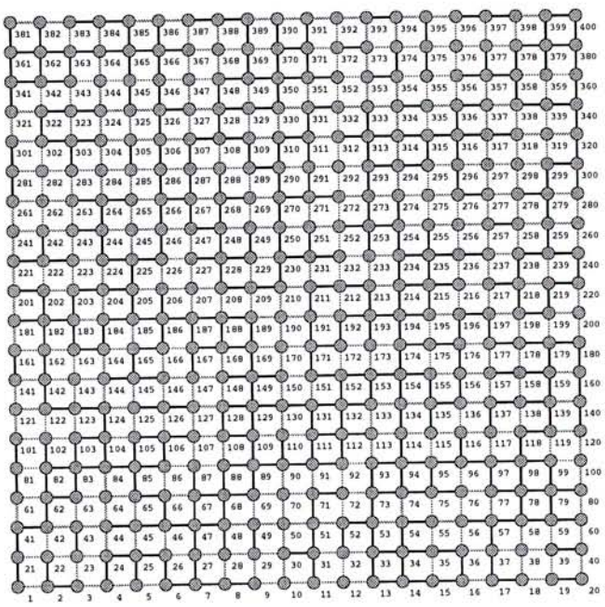


Figure 4.1: The configuration of a percolation network with $20 \times 20$ sites

$\varepsilon_x$ and $\varepsilon_y$ along the $x$-axis and $y$-axis are defined as [33]

$$\varepsilon_x = \frac{\Delta u}{\Delta x}, \quad \text{and} \quad \varepsilon_y = \frac{\Delta v}{\Delta y}, \tag{4.2}$$

In this model, the links (rods) are of negligible thickness. Therefore, we can ignore the strain along the perpendicular direction of the link. When we apply a stress $\sigma$ along the direction of a link of negligible thickness, we can obtain the following stress and strain $\varepsilon$ relation:

$$\varepsilon = \frac{\sigma}{E_h}, \tag{4.3}$$

where $h = 1, 2$ and $E$ is the Young's modulus of the thin link. The subscript $h$ means

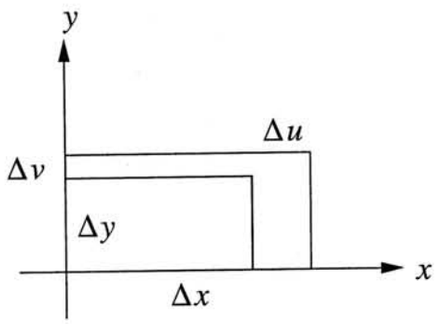



Figure 4.2: Normal strain and displacement relations.

that the corresponding physical quantity belongs to the bonds with relatively high thermal-expansion-coefficient if $h = 1$; otherwise, it belongs to the bonds with relatively low thermal-expansion-coefficient. The $x$-component $\sigma_{ij_x}$ and $y$-component $\sigma_{ij_y}$ of a stress acting on the node $i$ by the link $j$ are represented by

$$\sigma_{ij_x} = E_{ijh}\varepsilon_{ij}\cos\theta, \quad \text{and} \quad \sigma_{ij_y} = E_{ijh}\varepsilon_{ij}\sin\theta, \tag{4.4}$$

where $j = 1,...,4$, $i = 1,...,n$ and $\theta$ is the acute angle between the link and $x$-axis. The subscript $i$ means the corresponding physical quantity is with respect to the $i$-th node. In Figure 4.1, the number of $i$ is shown besides the right bottom side of the corresponding node. The subscript $x$ and $y$ are to represent the $x$-component and $y$-component of the corresponding physical quantity respectively. In Figure 4.3, it shows the $i$-th node (the central node) and its nearest neighbors. The subscript $j$ can have value $1,2,3,4$ representing the top, bottom, left, right link respectively. If the nearest neighbor of a node does not exist (the nodes at the boundary), the corresponding value of $\sigma_{ij_x}$ and $\sigma_{ij_y}$ will become zero. For example, $i = 1, j = 2$, $\sigma_{12_x} = 0$ and $\sigma_{12_y} = 0$. The $x$-component and $y$-component of the net stress at the $i$-th node are

$$\sigma_{i_x} = \sum_{j=1}^{4} \sigma_{ij_x}, \quad \text{and} \quad \sigma_{i_y} = \sum_{j=1}^{4} \sigma_{ij_y}. \tag{4.5}$$

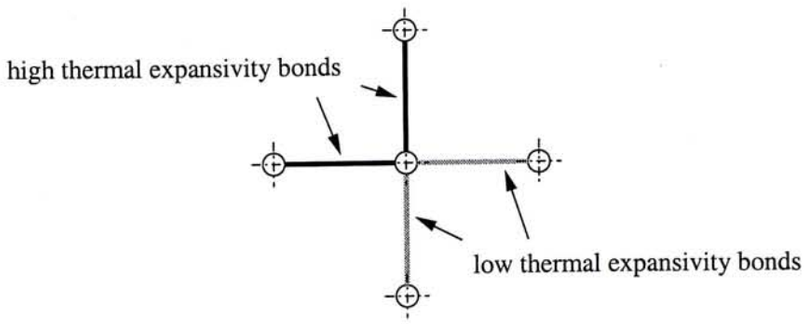



Figure 4.3: A site and its nearest neighbors.

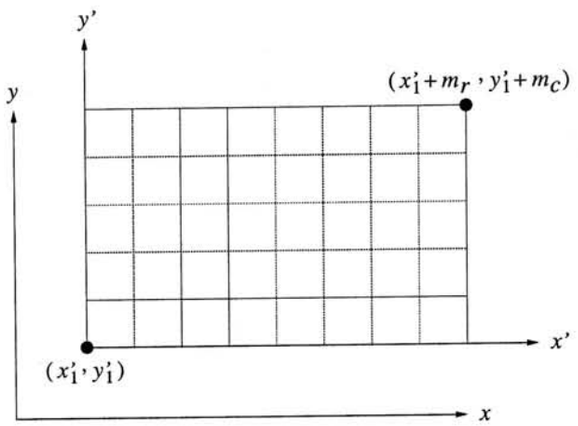



Figure 4.4: Coordinates transformation on calculating the thermal expansion coefficient.

For the ease of computation, we transform the $x$ and $y$ coordinates of the nodes by adding a suitable constant on the original coordinates as shown in Figure 4.4. The average value $\overline{x}_{\text{left}}$ of the leftmost nodes' $x$-coordinates and the average value $\overline{x}_{\text{right}}$ of the rightmost nodes' $x$-coordinates are defined as

$$\overline{x}_{\text{left}} = \frac{1}{m_c}\sum_{L=1}^{m_c} x_L, \quad \text{and} \quad \overline{x}_{\text{right}} = \frac{1}{m_c}\sum_{R=1}^{m_c} x_R. \tag{4.6}$$

The average value $\overline{y}_{\text{top}}$ of the topmost nodes' $y$-coordinates and the average value $\overline{y}_{\text{bottom}}$ of

the bottommost nodes' $y$-coordinates are defined as

$$\bar{y}_{\text{top}} = \frac{1}{m_r}\sum_{T=1}^{m_r} y_T, \quad \text{and} \quad \bar{y}_{\text{bottom}} = \frac{1}{m_r}\sum_{B=1}^{m_r} y_B, \tag{4.7}$$

where $x_L, x_R$ are the $x$-coordinates of the leftmost nodes and the rightmost nodes respectively; $y_T, y_B$ are the $y$-coordinates of the topmost nodes and the bottommost nodes respectively. The change of the length $\Delta l$ along $x$-axis and $y$-axis are

$$\Delta l_x = \bar{x}_{\text{right}} - \bar{x}_{\text{left}}, \quad \text{and} \quad \Delta l_y = \bar{y}_{\text{top}} - \bar{y}_{\text{bottom}}. \tag{4.8}$$

Since this is a model for an isotropic material and $m_c = m_r = \sqrt{n}$,

$$\Delta l = \frac{\Delta l_x + \Delta l_y}{2} - l. \tag{4.9}$$

Substitute $\Delta l$, $l$ and the change of the temperature into Equation (4.1), we will get the value of the effective thermal expansion coefficient.

According to the theory of elasticity [27], only three independent physical properties are used to describe these kind of physical systems. In [11], Albrecht *et al.* suggest a method to calculate the other two physical quantities — Young's modulus $E$ and Poisson's ratio $\nu$. They apply an external force on the boundary of the networks as shown in Figure 4.5. The $x$-component and $y$-component of the average net stress $\bar{\sigma}_x$ and $\bar{\sigma}_y$ are calculated at first. Finally, $\bar{\varepsilon}_x$ and $\bar{\varepsilon}_y$ are also computed. The Young's modulus and Poisson's ratio of the network are then calculated by the following relations

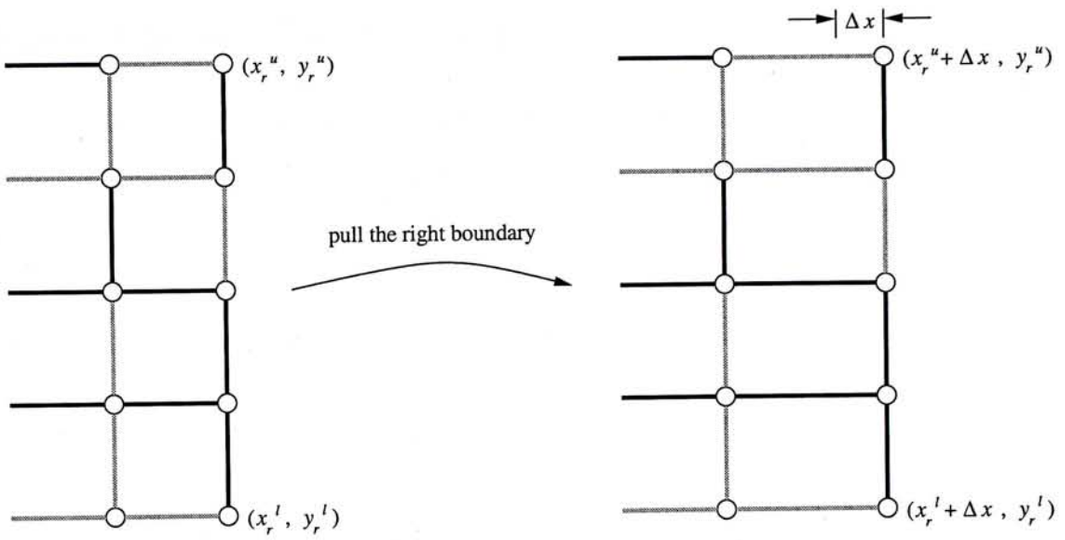



Figure 4.5: External force simulation.

$$E = \frac{\overline{\sigma}_x{}^2 - \overline{\sigma}_y{}^2}{\overline{\sigma}_x\overline{\varepsilon}_x - \overline{\sigma}_y\overline{\varepsilon}_y}, \quad \text{and} \quad \nu = \frac{\overline{\sigma}_x\overline{\varepsilon}_x - \overline{\sigma}_y\overline{\varepsilon}_y}{\overline{\sigma}_x\overline{\varepsilon}_y - \overline{\sigma}_y\overline{\varepsilon}_x}. \tag{4.10}$$

The derivations of $E$ and $\nu$ are shown in Appendix B.

### 4.2.2 Objective Function and Neighborhood Relation

Since our objective is to reduce the internal stress inside the percolation network due to the change of the temperature, we introduce the following objective function:

$$\mathcal{E} = \frac{1}{n}\sum_{i=1}^{n} |\sigma_{i_r}(\varepsilon_i)|, \tag{4.11}$$

where $|\sigma_{i_r}| = \sqrt{\sigma_{i_x}{}^2 + \sigma_{i_y}{}^2}$. Finally, the configuration of the percolation network is said to be stable if $\mathcal{E} \to c$, where $c$ is a constant and is close to zero. The final magnitude of the total resultant stress is not zero due to the boundary effect of the network, so $c$ does not

equal to zero. If the value of $n$ is larger, $c$ will become closer to zero.

In a real situation, the network will expand in continuous space. However, in a digital computer, it uses finite number of bits to represent a floating-point number. Hence, the precision of a floating-point number is limited. For this reason, we can imagine that we put the network on square grids. We represent the coordinates of the nodes on a grid point. The grid point to grid point distance is $g$. This method is acceptable if the original length $d$ of a link is much greater than $g$. During the expansion of the network, the nodes move between the grid points.

Initially, the lengths of the links are $d$. When we increase the temperature from $T_0$ to $T$ as shown in Figure 4.6, the lengths of the links are allowed to extend. Hence, some internal stresses will stay in the network. We define the initial configuration $P_{\text{init}}$ at this stage. Let the set of configurations that can be reached from $P_{\text{init}}$ by a sequence of moves of the nodes without any violation of the law of physics be $C_n$. The size $|C_n|$ of the configuration space is roughly bounded by

$$|C_n| < K \times (K-1) \times \ldots \times [K-(n-1)], \tag{4.12}$$

where $K$ is the total number of grid points in the possible region that the nodes can move. Our objective is to find a configuration, such that it has the minimum internal stress among all possible configurations.

## 4.3 Algorithm

Simulated annealing (SA) was introduced by Kirkpatrick *et al.* [12], and by Cerny [13] as an efficient stochastic algorithm for solving combinatorial optimization problems, such as the traveling salesman problem. Their development of the SA algorithms is based on the principle of Metropolis algorithm.

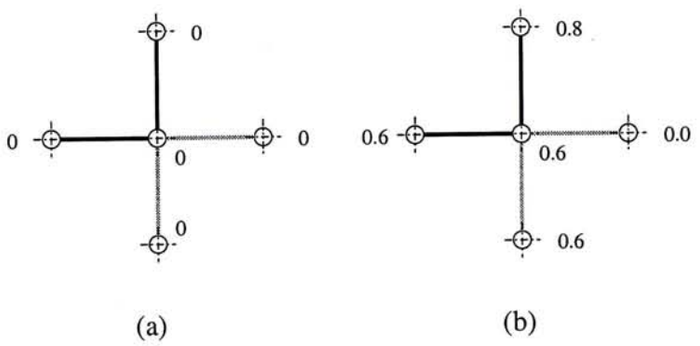



Figure 4.6: Stress distribution on a site and its nearest neighbors (a) at temperature $T_0$, (b) at temperature $T$.

### 4.3.1 Parallel Simulated Annealing

Traditionally, SA is hard to parallelize [34], but the nature of our problem enables us to do so. There are many approaches that are in detail in [15, 34]. Most of the parallel algorithms highly depend on the architecture of the machines. To accelerate the computational time of the simulations, we have to consider parallelizing simulated annealing algorithm on a particular type of parallel machine.

Basically, we use a shared memory model to implement our program. In this project, we use a parallel workstation Sun SPARCcenter 2000 that has 20 processors. We apply the multithreaded programming to do the program implementation. At each partitioning region, ideally, each CPU is mainly to calculate the local forces of the nodes. Less communication between the regions is required.

Since the algorithm is highly CPU-bound and not too much I/O is to be done, one lightweight processes (LWP) per CPU is enough to cover all simultaneous blocking system calls is called for [35]. In other words, to maximize processing power, the number of partitioning regions should be equal to the number of CPUs, such that the number of nodes at each region is equal. To minimize the communication overhead between the CPUs, we have to select a suitable partitioning method, so that the number of nodes at region boundaries are minimized. For the ease of implementation, we have to select a regular

shape region. We would like to know which shape is the best for partitioning. We consider several shapes including an equilateral triangle, an isosceles right-angle triangle, and a square. For simplicity, we assume the nodes can be evenly covered by the region of the shapes. Let us consider an equilateral triangle first. Let $A$ be the area and $d$ be the length of a equilateral triangle. Then, we get

$$A = \frac{\sqrt{3}}{4}d^2. \tag{4.13}$$

The boundary of the triangle equals $4.55\sqrt{A}$. Similarly, the boundary of an isosceles right-angle triangle and a square with the same area $A$ equals $4.83\sqrt{A}$ and $4\sqrt{A}$ respectively. From this, we see that the square shape is the best for partitioning. If the partitioning regions cannot be divided into equally square regions, we should select a rectangle which the difference between its width and length is the smallest (Figure 4.7).

| | | | | |
|---|---|---|---|---|
| 16 | 17 | 18 | 19 | 20 |
| 11 | 12 | 13 | 14 | 15 |
| 6 | 7 | 8 | 9 | 10 |
| 1 | 2 | 3 | 4 | 5 |

Figure 4.7: The regular shapes of the 20 partitioning regions.

Given a neighborhood structure, simulated annealing can be viewed as an algorithm that continuously attempts to transform the current configuration into one of its neighbors. Let $P$ be the current configuration at the $N$-th time step and $P'$ be the proposed configuration at the $(N+1)$-th time step. We denote $\mathcal{N}_p$ as the collection of allowed configurations. Consider a single step of transition; the probability $G[P,P']$ of moving from $P$ to $P'$ is given

by

$$G[P,P'] = \begin{cases} \frac{1}{8n} & \text{if } P' \in \mathcal{N}_p, \\ 0 & \text{otherwise.} \end{cases} \tag{4.14}$$

The probability $A[P,P']$ of accepting $P'$ is

$$A[P,P'] = \begin{cases} 1 & \text{if } \mathcal{E}(P') - \mathcal{E}(P) \leq 0, \\ \exp[(\mathcal{E}(P) - \mathcal{E}(P'))/t] & \text{otherwise.} \end{cases} \tag{4.15}$$

where $t$ is a parameter, called temperature, that controls the acceptance-rejection probability. Therefore, $P'$ is accepted even $\mathcal{E}(P) < \mathcal{E}(P')$ if

$$\exp[(\mathcal{E}(P) - \mathcal{E}(P'))/t] \geq \eta, \tag{4.16}$$

where $\eta \in [0,1)$ is a uniformly distributed random number. The probability that the configuration $P$ transits to $P'$ is

$$\Pr\{P \to P'\} = \begin{cases} G[P,P']A[P,P'] & \text{if } P' \neq P, \\ 1 - \sum_{Q \neq P} G[P,Q]A[P,Q] & \text{if } P' = P. \end{cases} \tag{4.17}$$

Equation (4.17) means that the searching direction in space $S$ may not be in the direction of gradient descent. In other words, we get the lowest cost value of $\mathcal{E}$ without being trapped in local minima if the adjustment of the parameter $t$ (cooling rhythm) is appropriate. The probability of ending in a global minimum is approaching to one.

During the simulation, each thread computes the local forces of the nodes in its region. Therefore, parallel moves of the nodes are permitted. A node at the boundary of the region is locked during the update of its position. The value of $\mathcal{E}$ is then updated due to the 'local', computed value of each region. All of the computational process can be done asynchronously. Hence, the convergence of the algorithm will not be affected.

### 4.3.2 The Physical Annealing Schedule

In this section, we will discuss about the annealing schedule that we used in this problem. The simplest and most common method of the annealing schedule is temperature decrement rule

$$t_{N+1} = \gamma t_N, \tag{4.18}$$

where $\gamma$ is constant close to, but smaller than, 1. This exponential cooling scheme was first proposed by Kirkpatrick *et al.* [12] with $\gamma = 0.95$. Randelman and Grest [36] compared this strategy with a linear cooling scheme in which $t$ is reduced on every $L$ trials

$$t_{N+1} = t_N - \Delta t. \tag{4.19}$$

They found the reductions achieved using the two schemes to be comparable, and also noted that the final value of $\mathcal{E}$ was, in general, improved with slower cooling rates, at the expense of greater computational effort. Finally, they observed that the algorithm performance depended on the cooling rate $\Delta t/L$ rather than on the individual values of $\Delta t$ and $L$. Many researchers have proposed more elaborate annealing schedules, most of which are in some respect adaptive, using statistical measures of the algorithm's current performance to modify its control parameters. These are well reviewed by van Laarhoven and Aarts [14].

Ideally, we would like a method for determining the annealing schedule based on the behavior of the specific system. We obtain some guiding principles by considering the statistical mechanics of the problem. One good rule of thumb is that we would like to minimize the variation from equilibrium. During the annealing, we can measure the energy (cost) and specific heat (variance of the cost) at each temperature value. These values can be used to help determine the annealing schedule. However, since we are constantly changing $t$, it is not easy to obtain equilibrium values of $\mathcal{E}$ and $C_H$ for each temperature. But, we can estimate the values.

In our project, we are concerned with the solution quality rather than the run-time of the simulation, so we apply the physical annealing schedule. We denote $t_k$ as the range of temperatures and $N_k$ as the number of Metropolis iterations at each $t_k$. We would like to choose $N_k$, so thermal equilibrium is reached at $t_k$ before moving to $t_{k+1}$. We can check this using standard methods for testing for equilibration. Or alternatively, we assume $\mathcal{E} = \exp(\mathcal{L}_k)$ and $\tau_{\exp}$ is the value of intersection point between the $x$-axis of $\mathcal{L}_k$ and the tangent line of $\exp(\mathcal{L}_k)$ at $\mathcal{L}_k = 0$, where $\mathcal{L}_k \in [0, N_k - 1]$. Then we can estimate $\tau_{\exp}$ from measurements of $\mathcal{E}$ at smaller values of $\mathcal{L}$, and set $N_k \approx \tau_{\exp}$. Note that we should not set $N_k \geq 4\tau_{\exp}$ because fluctuations of $\mathcal{E}$ exist.

$$C_{H_k} = \frac{d\mathcal{E}}{dt} \approx \frac{\Delta\mathcal{E}_k}{\Delta t_k} \tag{4.20}$$

$$t_{k+1} \approx t_k - \frac{\Delta\mathcal{E}_{k+1}}{C_{H_k}} \tag{4.21}$$

By a suitable initial value of $\Delta t_0$ at $k = 0$ as shown in the previous section, we can minimize the objective function through this simulated annealing algorithm.

## 4.4 Results

We have implemented the algorithms on a parallel 20-processor Sun Sparc2000/20 workstation that is running Solaris 2.6 operation system. The programming model, which is used in this implementation, is the peer mode [34, 37]. All the pseudo-random numbers are generated by using the well-known combined linear congruential algorithm and 32-bit integer arithmetic (Appendix C). Each thread has its pseudo-random number generator that is initialized by another well-known 48-bit integer arithmetic pseudo-random number.

The networks are divided by 20 regions equally and each thread has its responsibility to do the computation of each local region. Since we can update all sites at the same time, this would not violate detailed balance. But, this fact will not true at nearly the final time. We set a threshold value $t_{\text{threshold}}$ to destroy all the threads when $t$ is reduced to $t_{\text{threshold}}$ and a serial process is then to be continued the computation. The run-time for the part of the serial computation is much less than that of parallel computation. On the whole, we can ignore the run-time for the part of the serial computation.

We have performed the simulation on a percolation network with 400 nodes and 900 nodes. The initial configuration of a two-component percolation network with $20 \times 20$ sites is shown in Figure 4.8. The material of $\alpha_1$ bond is epoxy; another one is carbon strip. The parameters are: $p = 0.5$, $E_1 = 34\,\text{GPa}$, $E_2 = 260\,\text{GPa}$, $E = 120\,\text{GPa}$, $\nu = 0.13$, $\alpha_1 \Delta T = 2.6 \times 10^{-4}$, $\alpha_2 \Delta T = 3.8 \times 10^{-5}$, $d = 5$. The computed $\alpha \Delta T$ value is $1.2 \times 10^{-4}$. The initial configuration of a two-component percolation network with $20 \times 20$ sites is shown in Figure 4.8. The resultant stress on a node is represented by uniform gray scale color. Deeper color means higher resultant stress on a node. The scaled value of internal stress of a node is shown on its right bottom corner. Note that the actual size of the nodes is much smaller than that shown in the figure. In Figure 4.9, it shows the final configuration of a two-component percolation network with $20 \times 20$ sites. It can be seen that the resultant stress on each node is very small. The comparison of the serial, and parallel run-time of this algorithm is shown in Table 4.1.

Table 4.1: Comparisons between the network with 400 nodes and with 900 nodes.

| Number of nodes | 400 | 900 |
|---|---|---|
| Average number of nodes per processor | 20 | 45 |
| Approximate parallel run-time / min | 80 | 130 |
| Approximate serial run-time / min | 480 | 920 |
| Approximate speed-up | 6 | 7 |

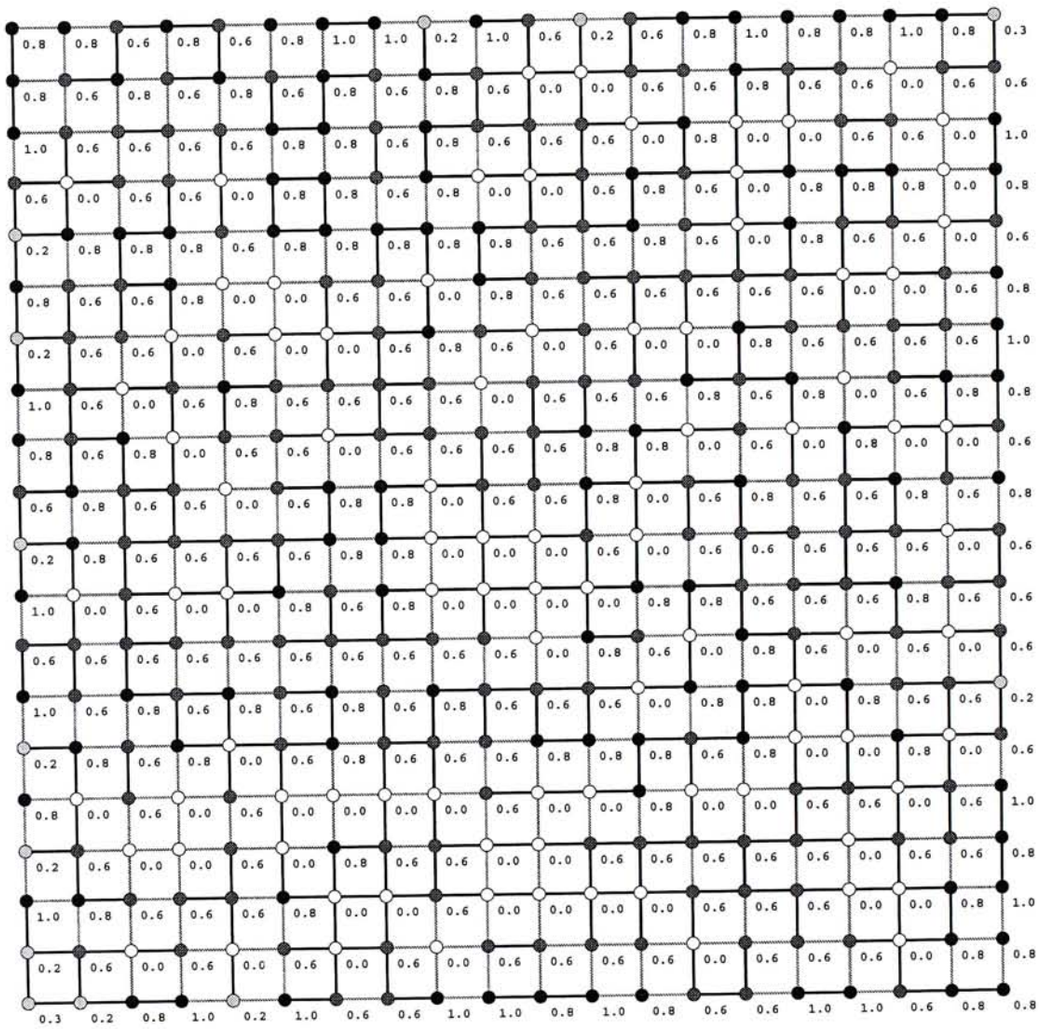


Figure 4.8: A initial configuration of a two-component percolation network with $20 \times 20$ sites.

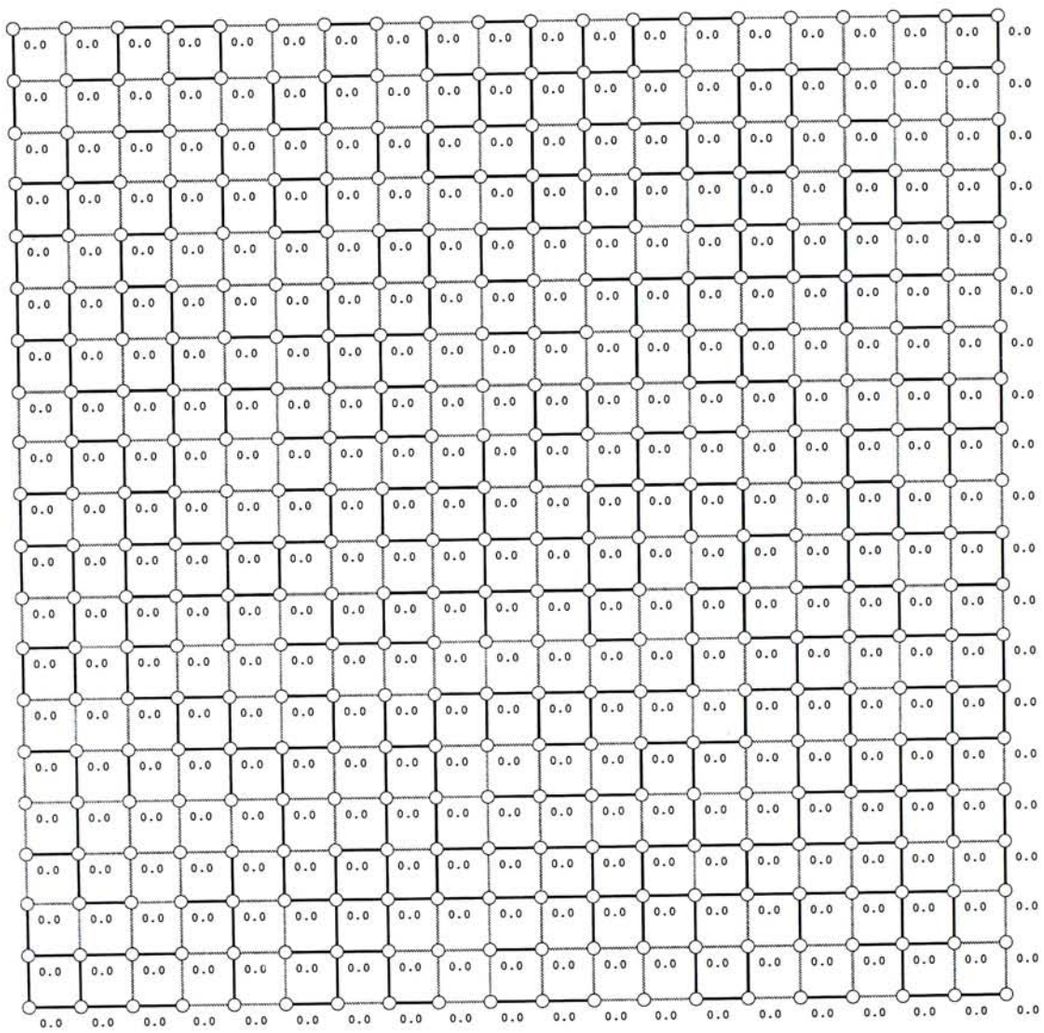


Figure 4.9: Final configuration of a two-component percolation network with $20 \times 20$ sites.

## 4.5 Discussion

We have performed stochastic simulations of a two-dimensional percolation network with mixed relatively high thermal-expansion-coefficient and relatively low thermal-expansion-coefficient bonds, based on simulated annealing-based procedure with the physical-annealing schedule. We would like to point out that this algorithm can be used to calculate the effective thermal expansion coefficient of composite materials even the original thermal expansion coefficients of a matrix and a filler are not extreme. Through this simulation, we show the universal application of this algorithm and model on solving this kind of problem. Importantly, we need not assume that the thermal expansion coefficients of a matrix and a filler have linear relation between the temperature in our formulation. We can solve a problem even they have non-linear relations. Also, the formulation of algorithm and model has a potential to calculate the physical properties of a system with short fiber as the filler.

In our early consideration, we only concentrate the materials with positive thermal expansion coefficient. But, there exists some materials have negative thermal expansion coefficient, that is the length of the material will shrink with the increase of the temperature. If we mix these two kinds (positive and negative $\alpha$) of materials, the overall thermal expansion coefficient of the composite material may become nearly zero. This kind of composite is very useful in engineering industry. Take a space shuttle as an example, the outer shell of a space shuttle consists of the composite materials with zero thermal expansion coefficient. This kind of material can be simulated using the same algorithm by changing the value of thermal expansion coefficient.

# Chapter 5

# Scaling Properties

In the previous chapter, we have calculated the thermal expansivity of percolation networks by a parallel simulated annealing algorithm. Recall that percolation theory and its concepts, originally applied to scalar transport problems, have been recently applied to a variety of vector transport problems. The so-called Elastic Percolation Networks (EPNs) consist of elastic elements (springs or beams) which can transport stress and can support displacement modes according to the microscopic force law (Hooke's law) being used to describe the interaction between sites and bonds of the network. Many interesting features of EPNs such as their critical behavior and their use in modeling vectorial phase transitions have already been studied in 2D and 3D. It is evident that removal of a single bond sometimes causes a large reduction in the elastic constant (macroscopic property) of the system.

Scaling is the character of the properties of composite materials that change abruptly. To our knowledge, there are many studies on scaling behaviors of physical properties of percolation networks [28, 31, 38, 39, 40], but the study on scaling properties of the thermal expansion coefficient has not been found. In this chapter, we will study scaling behavior of the thermal expansion coefficient of a percolation network.

## 5.1 Problem Define

In [38], Bergman *et al.* found that the critical behavior of the elastic moduli differs from that of the electrical conductivity $\sigma$ in the same models. We are interested in knowing the existence of the scaling behavior of thermal expansion coefficient $\alpha$.

Consider the un-deformed state of a body in the absence of external forces at some given temperature $T_0$. If the body is at a temperature $T$ different from $T_0$, then, even if there are no external forces, it will in general be deformed, on account of thermal expansion. In free thermal expansion of the body (external forces being absent), there can be no internal stress. Then, we get Equation (4.1).

Suppose that in a percolation network a fraction $p$ of bonds represent links (rods) of negligible thickness with a thermal expansion coefficient $\alpha_1$ and the remaining bonds are links of negligible thickness with a thermal expansion coefficient $\alpha_2$. Both $\alpha_1$ and $\alpha_2$ can be selected from any probability density function $H(\alpha)$. Let us consider the simplest case

$$H(\alpha) = p\delta(\alpha-\alpha_1)+(1-p)\delta(\alpha-\alpha_2), \tag{5.1}$$

i.e., $\alpha$ takes the values $\alpha_1$ and $\alpha_2$ with probability $p$ and $1-p$, respectively. We can consider a more general distribution,

$$H(\alpha) = p\mathrm{f}_1(\alpha)+(1-p)\mathrm{f}_2(\alpha), \tag{5.2}$$

where $\mathrm{f}_1$ and $\mathrm{f}_2$ are two continuous and normalized probability density functions. But, in this chapter, we only consider the case shown in Equation (5.1). We would like to know the value of exponent $f$ if the thermal expansion coefficient obeys the following scaling law

$$\alpha \sim (p-p_c)^f, \tag{5.3}$$

where $p_c$ is a percolation threshold.

## 5.2 Physical Model

In the literature [28, 29, 30], a percolation network with the underlying hexagonal-like network structure was studied to calculate its elastic properties. In [31, 32], not only were the bond-stretching forces calculated, but also the angle-bending forces were also calculated in the network. But, for simplicity, we do not consider the angle-bending forces in our model. To study scaling properties of thermal expansivity of a percolation network, we need a model, which has some difference as shown in Chapter 4, to perform simulations. But, the terminology and the symbols have the same meaning in Chapter 4 except that we redefine the meaning of them in this chapter.

### 5.2.1 The Physical Properties

Again, consider a finite lattice for an $m_r \times m_c$ percolation network and the total number of sites be $n$. The network consists of bonds chosen randomly to have a thermal expansion coefficient $\alpha_1$ and a force constant $\beta_1$ with probability $p$ or $\alpha_2$ and $\beta_2$ with probability $q$, where $p+q=1$. Without loss of generality, we choose $\alpha_1 > \alpha_2$. $m_r$ and $m_c$ have the same value $\sqrt{n}$, where $\sqrt{n}$ are positive integers.

### 5.2.2 Bond Stretching Force

In this model of percolation networks, the links (rods) are of negligible thickness. Therefore, we can ignore the strain along the perpendicular direction of the link. When we apply a stress $\sigma$ along the direction of a link, we can obtain the following stress and strain $\varepsilon$ relation:

$$\varepsilon = \frac{\sigma}{\beta_h}. \tag{5.4}$$

where $\beta$ is the stretching force constant of the thin link; $h$ can be 1 or 2. The $x$-component $\sigma_{ij_x}$ and $y$-component $\sigma_{ij_y}$ of a stress acting on the node $i$ by the link $j$ are represented by

$$\sigma_{ij_x} = k_{ijh}\varepsilon_{ij}\cos\theta \quad \text{and} \quad \sigma_{ij_y} = k_{ijh}\varepsilon_{ij}\sin\theta, \tag{5.5}$$

where $j = 1,...,4, i = 1,...,n$ and $\theta$ is the acute angle between the link and $x$-axis. The subscript $i$ denotes the physical quantities of the $i$-th node. The subscript $x$ and $y$ is to represent the $x$-component and $y$-component of the corresponding physical quantity respectively. The subscript $j$ can have value $1,2,3,4$ representing the top, bottom, left, right link respectively. If the nearest neighbor of a node does not exist (the nodes at the boundary), the corresponding value of $\sigma_{ij_x}$ and $\sigma_{ij_y}$ will become zero. Then, the $x$-component and $y$-component of the net stress at the $i$-th node are obtained from Equation (4.5). Finally, we obtain the value of $\Delta l$ from Equation (4.9), $E$ and $\nu$ from Equation (4.10).

### 5.2.3 Objective Function and Configuration Space

Since our objective is to reduce the internal stress inside the percolation network due to the change of the temperature, we introduce the following objective function:

$$\mathcal{E} = \frac{1}{n}\sum_{i=1}^{n} |\sigma_{i_r}(\varepsilon_i)|, \tag{5.6}$$

where $|\sigma_{i_r}| = \sqrt{{\sigma_{i_x}}^2 + {\sigma_{i_y}}^2}$. The final configuration of the percolation network is said to be stable if $\mathcal{E} \to c$, where $c$ is a constant and is close to zero. The final magnitude of the total resultant stress is not zero due to the boundary effect of the network, so $c$ does not equal to zero. If the value of $n$ is larger, $c$ will become closer to zero. Again, a grid point model is used as described in Chapter 4.

Initially, the lengths of the links are $d$. When we increase the temperature from $T_0$ to $T$, the lengths of the links are not allowed to extend. Hence, some internal stresses will stay in the links. We define the initial configuration $P_{\text{init}}$ at this stage. Let the set of configurations that can be reached from $P_{\text{init}}$ by a sequence of moves of the nodes without any violation of the law of physics be $C_n$. The size $|C_n|$ of the configuration space is roughly bounded by

$$|C_n| < K \times (K-1) \times ... \times [K-(n-1)], \tag{5.7}$$

where $K$ is the total number of grid points in the possible region within which the nodes can move. Although we will introduce a new value of $P_{\text{init}}$ in a later section, this bound will not be changed. Our objective is to find a configuration, such that the configuration has minimum internal stress among the entire possible configurations.

## 5.3 Algorithm

The new algorithms proposed in this chapter are also simulated annealing (SA) based. There is much work on the improvement of the run-time of SA. In this section, we will propose an algorithm that is different from the existing work and is based on another point of view.

### 5.3.1 Simulated Annealing

To accelerate the computational speed, we have proposed a parallel SA algorithm to perform the simulations. We would like to have a further improvement on the speed for this kind of problem. Since the rate of convergence is sensitive to the initial condition of algorithm, we use a conjectural method to initialize the initial values of nodes. Hence, the number of steps from $P_{\text{init}}$to $P_{\text{opt}}$ is reduced.

Given a neighborhood structure, simulated annealing can be viewed as an algorithm that continuously attempts to transform the current configuration into one of its neighbors. Let $P$ be the current configuration at the $N$-th time step and $P'$ be the proposed configuration at the $(N+1)$-th time step. We denote $\mathcal{N}_p$ as the collection of allowed configurations. Consider a single step of transition; the probability $G[P,P']$ of moving from $P$ to $P'$ is given by

$$G[P,P'] = \begin{cases} \frac{1}{8n} & \text{if } P' \in \mathcal{N}_p, \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

The probability $A[P,P']$ of accepting $P'$ is

$$A[P,P'] = \begin{cases} 1 & \text{if } \mathcal{E}(P') - \mathcal{E}(P) \leq 0, \\ \exp[(\mathcal{E}(P) - \mathcal{E}(P'))/t] & \text{otherwise.} \end{cases} \tag{5.9}$$

where $t$ is a parameter, called temperature, that controls the acceptance-rejection probability. Therefore, $P'$ is accepted even $\mathcal{E}(P) < \mathcal{E}(P')$ if

$$\exp[(\mathcal{E}(P) - \mathcal{E}(P'))/t] \geq \eta, \tag{5.10}$$

where $\eta \in [0,1)$ is a uniformly distributed random number. The probability that the configuration $P$ transits to $P'$ is

$$\Pr\{P \to P'\} = \begin{cases} G[P,P']A[P,P'] & \text{if } P' \neq P, \\ 1 - \sum_{Q \neq P} G[P,Q]A[P,Q] & \text{if } P' = P. \end{cases} \tag{5.11}$$

There are several methods to accelerate the run-time of the algorithm detailed by [14]. The proposed methods are to change the probability density of $G[P,P']$ and $A[P,P']$. For example, the probability density functions are Gaussian, Lorentzian distribution. However, these methods will affect the result quality of the simulation. Therefore, we will not use these methods.

### 5.3.2 The Conjectural Method

Consider a sequence of successful moves of configurations from initial configuration $P_0$ into the optimal configuration $P_{opt}$. The sequence can be represented as

$$P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow ... \rightarrow P_{opt}. \tag{5.12}$$

The symbol $\rightarrow$ means the transition from one configuration to another. We now define a meta configuration $P_{meta}$ which can be a configuration between $P_0$ and $P_{opt}$. The next question is how to estimate $P_{meta}$ in our problem.

Let us recall the algorithm described in Chapter 4. Initially, the lengths of the links in a percolation network are $d$. When we increase the temperature from $T_0$ to $T$, the lengths of the links are not allowed to extend. Hence, the internal stresses will stay in the network. This is the initial configuration $P_{init}$ of the network at this stage. Without loss of generality, we select $\alpha_1 < \alpha_2$. We denote $P_{max}$ as a configuration of a percolation network of which links have their lengths $d + \alpha_2 T d$ at temperature $T$. From our experience, we should set the initial lengths of the links at $d + (\frac{\alpha_1+\alpha_2}{3})Td$ instead of $d$. This is one of possible configurations of $P_{meta}$. We should not be too "greedy"; otherwise the transitions cannot convert to $P_{opt}$. For example, $d + (\frac{\alpha_1+\alpha_2}{2})Td$ is not a good choice of $P_{meta}$. It is because the "perturbation" we are using is not large enough to move the state out of the local maxima. The result of a simulation with "over conjectural" initial condition is shown in Figure 5.1. From the figure, we can see that there exist some stresses (represented by black dots) staying on the boundaries of the network due to the "over conjectural" initial condition.

In short, a conjectural simulated annealing algorithm that is an SA based algorithm starting with reasonable conjectural initial values of a configuration can save a lot of computational time on the initial process. From our observation, our conjectural condition is very robust: it works for problems with different link lengths and different number of sites (nodes).

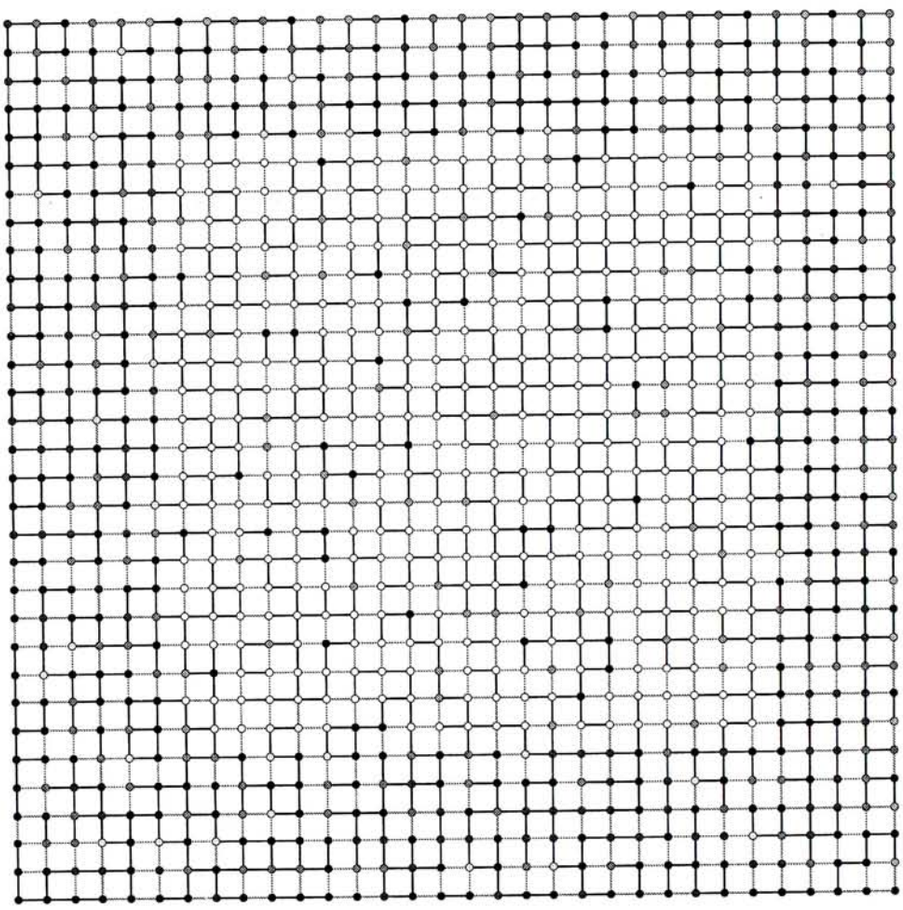


Figure 5.1: The result of a simulation with "over conjectural" initial condition.

### 5.3.3 The Physical Annealing Schedule

In [14], it reviewed several types of annealing schedules. A fixed annealing schedule means the annealing temperature $t$ is a function of some annealing parameters, e.g., the number of iterations of the program, and the previous value of annealing temperature $t$. If we apply a fixed annealing schedule to our problem, we will need at least four sets of annealing parameters, namely, for a serial SA, a parallel SA, a conjectural SA, and a conjectural parallel SA, to perform the simulation. From this, we can see the advantage of the physical annealing schedule as described in Chapter 4. The physical annealing schedule will reduce the annealing temperature adaptively according to the estimation of criteria. Also, the system can reach equilibrium at each intermediate stage before decreasing annealing temperature. Therefore, relative to fixed cooling schedules, high quality results are obtained from our simulations.

Again, we apply the physical annealing schedule. During the annealing, we can measure the energy (cost) and specific heat (variance of the cost) at each temperature value. These values can be used to help determine the annealing schedule. However, since we are constantly changing $T$, it is not easy to obtain equilibrium values of $\mathcal{E}$ and $C_H$ for each temperature. But, we can estimate the values.

We denote $t_k$ as the range of temperatures and $N_k$ as the number of Metropolis iterations at each $t_k$. We would like to choose $N_k$, so that thermal equilibrium is reached at $t_k$ before moving to $t_{k+1}$. We can check this using standard methods for testing for equilibrium. Or alternatively, we assume $\mathcal{E} = \exp(\mathcal{L}_k)$ and $\tau_{\exp}$ is the value of intersection point between the $x$-axis of $\mathcal{L}_k$ and the tangent line of $\exp(\mathcal{L}_k)$ at $\mathcal{L}_k = 0$, where $\mathcal{L}_k \in [0, N_k - 1]$. Then we can estimate $\tau_{\exp}$ from measurements of $\mathcal{E}$ at smaller values of $\mathcal{L}$, and set $N_k \approx \tau_{\exp}$. Note that we should not set $N_k \geq 4\tau_{\exp}$ because fluctuations of $\mathcal{E}$ exist.

$$C_{H_k} = \frac{d\mathcal{E}}{dt} \approx \frac{\Delta\mathcal{E}_k}{\Delta t_k} \tag{5.13}$$

$$t_{k+1} \approx t_k - \frac{\Delta \mathcal{E}_{k+1}}{C_{H_k}} \tag{5.14}$$

By a suitable initial value of $\Delta t_0$ at $k = 0$ as shown in the previous section, we can minimize the objective function through this simulated annealing algorithm.

As described in Chapter 4, we apply multithreaded programming in the implementation. During the simulation, each thread computes the local forces of the nodes in its region. Therefore, parallel moves of the nodes are permitted. A node at the boundary of the region is locked during the update of its position. The value of $\mathcal{E}$ is then updated due to the 'local', computed value of each region. All of the computational process can be done asynchronously. Since the conjectural method is to give a suitable initial condition of the network, the convergence of the algorithm will not be affected.

## 5.4 Results

We have implemented the algorithm on a Sun SPARCcenter 2000 workstation that is running Solaris 2.6 operation system with 20 processors. The programming model is the same as described in Chapter 4. All the pseudo-random numbers are generated by using the well-known combined linear congruential algorithm and 32-bit integer arithmetic. Each thread has its pseudo-random number generator.

The networks are divided by 20 regions equally and each thread has its responsibility to do the computation of each local region. Each node will only appear in the corresponding thread. Since we can update all sites at the same time, this would not violate detailed balance. However, this fact will not be true at near the final stage. We set a threshold value $t_{\text{threshold}}$ to destroy all the threads when $t$ is reduced to $t_{\text{threshold}}$ and a serial process is then to be continued the computation. The run-time for the part of the serial computation is much less than that of the parallel computation. On the whole, we can ignore the run-time for the part of the serial computation.

The final configuration of a two-component network with $32 \times 32$ sites is shown in Figure 5.2. The resultant stress on a node is represented by uniform grey scale color. Deeper color means higher resultant stress on a node. The scaled value of internal stress of a node is shown in the figure. It can be seen that the resultant stress on each node is very small.

We have performed the simulation on a percolation network with 1024, 1225, and 1600 nodes. The CPU time of the networks is shown in Table 5.1. We compare the parallel SA's run-time with the SA's run-time to obtain the speed-up of the algorithm. To simulate a network with 1600 nodes, the speed-up is about 8.5. It is close to half the number of processors.

In Table 5.2, it shows the run-time of SA, parallel SA, conjectural SA, and the conjectural parallel SA on a simulation of the network with 1024 nodes. The run-time of these methods are compared with SA algorithm. The speed-up of the algorithm to simulate the network is about 9. If we increase the number of nodes to 1225 or 1600, the difference of the conjectural parallel SA's speed-up and the parallel SA's speed-up does not increase too much. The difference is still about 2. In the case of the conjectural SA's speed-up and SA's speed-up, the difference increases gradually. The difference is about 4. From these, we can see that the conjectural method does not seem to depend on the number of nodes too much. It is understandable because we change the initial configuration of a network and the conjectural value is independent of the number of nodes.

Table 5.1: The CPU time of the networks with 1024, 1225, and 1600 nodes.

| Number of nodes | 1024 | 1225 | 1600 |
|---|---|---|---|
| Average number of nodes per processor | 51 | 61 | 80 |
| Approximate parallel run-time / min | 155 | 175 | 235 |
| Approximate serial run-time / min | 1130 | 1395 | 2000 |
| Approximate speed-up | 7 | 8 | 8.5 |

To investigate the existence of the critical properties of the thermoelastic moduli of random thermal expansion coefficient networks near the threshold $p_c$, we study the problem in several cases. We consider three cases of the ratio of $\alpha_1$ and $\alpha_2$ and the ratio of $\beta_1$ and

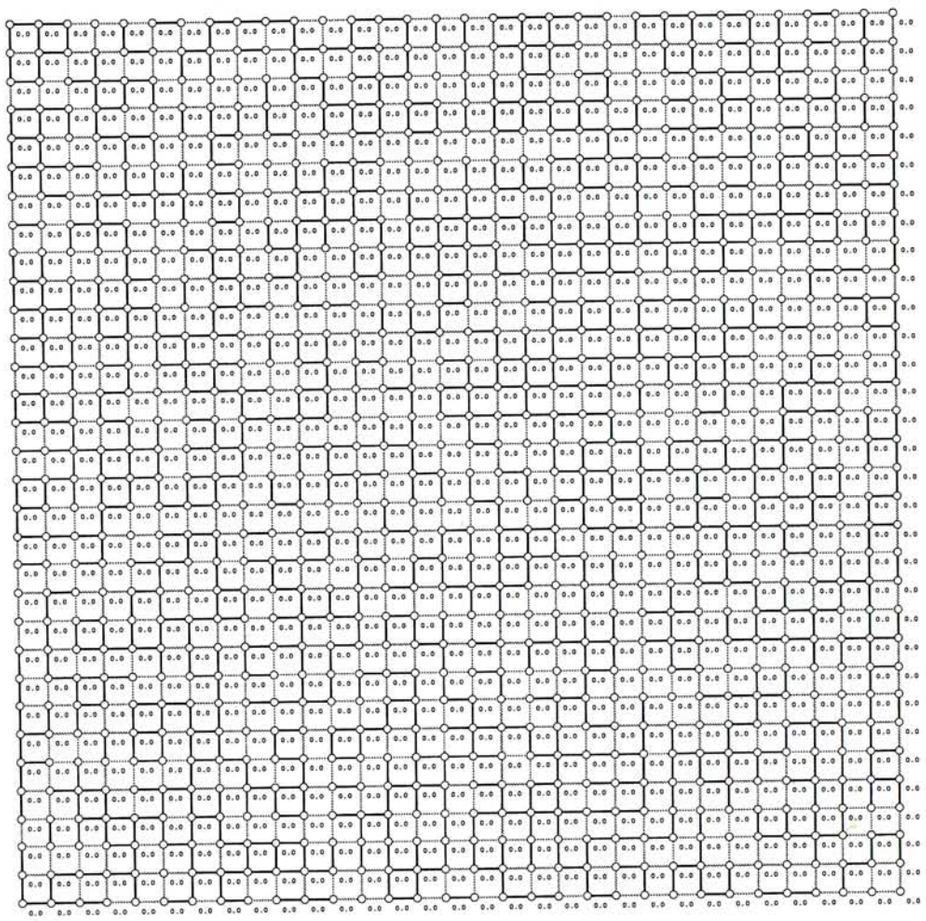


Figure 5.2: Final configuration of a two-component percolation network with $32 \times 32$ sites.

$\beta_2$. We use the ratio values instead of the real data. It is because we are concerned with what the order-of-magnitude of the physical quantities causes the scaling behavior. The ratio values are required to fulfill the existing possibility in physical world. In the three cases of simulations, $p$ represents the fraction of $\alpha_2$ bonds. The number of sites is 1600.

### 5.4.1 Case I

In Case I, $\alpha_1/\alpha_2 = 0.01$; $\beta_1/\beta_2 = 10^{-7}$. The results of this case are shown in Table 5.3. No scaling behavior of the thermal expansion coefficient is obtained.

Table 5.2: The speed-up of the parallel SA, the conjectural SA, and the conjectural parallel SA on the computation of the network with 1600 nodes.

| Algorithm | SA | Parallel SA | Conjectural SA | Conjectural Parallel SA |
|---|---|---|---|---|
| Average number of nodes per processor | 1024 | 51 | 1024 | 51 |
| Approximate parallel run-time / min | - | 155 | - | 126 |
| Approximate serial run-time / min | 1130 | - | 283 | - |
| Approximate speed-up (comparing with SA) | 1 | 7 | 4 | 9 |

Table 5.3: Simulation results of Case I.

| $p$ | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
|---|---|---|---|---|---|
| $\alpha/\alpha_2$ | 0.05 | 0.19 | 0.42 | 0.62 | 0.87 |

### 5.4.2 Case II

In Case II, $\alpha_1/\alpha_2 = 100$; $\beta_1/\beta_2 = 10^7$. The results of this case are shown in Table 5.4. No scaling behavior of the thermal expansion coefficient is obtained.

Table 5.4: Simulation results of Case II.

| $p$ | 0.1 | 0.3 | 0.5 | 0.7 | 0.9 |
|---|---|---|---|---|---|
| $\alpha/\alpha_1$ | 0.85 | 0.59 | 0.40 | 0.17 | 0.03 |

### 5.4.3 Case III

In Case III, $\alpha_1/\alpha_2 = 0.01$; $\beta_1/\beta_2 = 10^7$. The results of this case are shown in Table 5.5. The scaling behavior of the thermal expansion coefficient of the percolation network is found. The estimated value of $f$ is about 1.2. From the table, the thermal properties of the network change abruptly. Since the order-of-magnitude of the ratio of stretching-force constant $\beta_1$ to $\beta_2$ is much larger than that of the ratio of thermal expansion coefficient, the behavior seems to be governed by the elastic properties of a percolation network.

Table 5.5: Simulation results of Case III.

| $p$ | 0.5 | 0.55 | 0.6 | 0.65 | 0.7 | 0.75 |
|---|---|---|---|---|---|---|
| $\alpha/\alpha_2$ | 0.0 | 0.063 | 0.14 | 0.24 | 0.33 | 0.43 |

## 5.5 Discussion

The critical properties of the thermoelastic moduli of random thermal expansion coefficient networks near the threshold $p_c$ are investigated by constructing a square network. The algorithm is tested by simulations of a series of two-dimensional percolation networks near the threshold $p_c$. Surprisingly, the preliminary simulation results show that the percolating phenomenon of the thermal expansion coefficient does exist under certain conditions. The behavior seems to be governed by the mechanical properties of a percolation network.

Also, we have proposed and implemented a conjectural parallel simulated annealing algorithm to accelerate the speed of computation. A conjectural simulated annealing algorithm that is an SA based algorithm starting with reasonable conjectural initial values of a configuration can prevent wasting the computational time on the initial process. Comparing with a serial simulated annealing algorithm, the speed-up of the algorithm to simulate a network with 1024 nodes is about 9, which is close to half the number of the processors, provided that a suitable conjectural method is used.

# Chapter 6

# Conclusion

We have performed stochastic simulations of two-dimensional square percolation networks by the physical annealing schedules. Our approach does not depend on the lattice structure so that it can be applied in a percolation network consisting of a triangular, square or hexagonal lattice. The change of a lattice structure will not affect our approach because the order-of-magnitude of the number of a site's nearest neighbors does not change. Apart from handling a network with a regular lattice shape, this approach can also handle a network with an irregular lattice shape. In this case, we only have to redefine the neighborhood structure and to keep the information of a site's nearest neighbors during simulations. Moreover, this approach has the potential to simulate a network formed by more than two materials. Hence, we can see the flexibility of this formulation.

In the project, the electrical conductivity and the thermal expansivity of percolation networks have been computed by simulated annealing based algorithms. Actually, we can study the other physical properties, say thermal conductivity, with similar formulation. It indicates the good applicability of our proposed method to calculate the physical properties of percolation networks.

Our algorithms perform a stochastic search for equilibrium configurations of networks.

The simulation is started by the impact of external perturbation. Then, the corresponding physical properties are obtained after simulations. The expected serial run-time for the physical annealing schedules is $O(n \ln^2 n \ln(ln\, n))$. The physical annealing schedules will reduce the annealing temperature adaptively according to the estimation of criteria. Hence, compared to fixed cooling schedules, higher quality results have been obtained from simulations because the system can reach the equilibrium at each intermediate stage before decreasing the annealing temperature. The results from the new conjectural parallel algorithm have shown that even for a large number of nodes, the program runs relatively fast. Compared with a serial SA algorithm, the speed-up is close to half the number of the processors.

To our knowledge, there are many studies on scaling properties of physical properties of percolation networks but the study on scaling properties of the thermal expansion coefficient has not been found. From our preliminary simulation results, they have shown that the thermal expansion coefficient of a percolation network has scaling behavior under certain conditions. The behavior seems to be governed by the mechanical properties of a percolation network.

In summary, we have proposed and implemented efficient simulated annealing based algorithms to compute the physical properties of percolation networks in this project. The proposed method is flexible, efficient and there is good applicability of this kind of problem.

# Appendix A

# An Example on Studying Electrical Resistivity

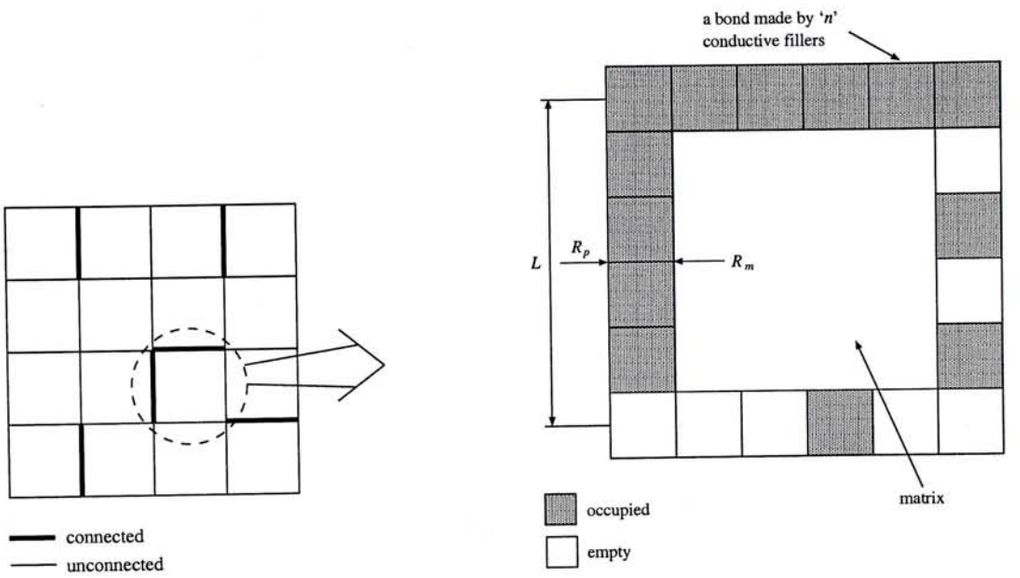
The original lattice structure model in [4] is a honeycomb structure. But, the structure in our case is a square structure. Owing to the case of understanding on this project, we use the idea in [4] to apply on a square structure mode (Figure A.1).

Consider a composite with the volume fraction of a filler, $v$, is calculated by using filling factor, $\phi$, depending on the lattice structure, and probability of particles occupying the grain boundaries (the sites), $p$, which is the ratio of the number of present particles to that of total particles occupied in the system. The relation is described as

$$v = \phi p. \tag{A.1}$$

The filling factor of a square structure, as shown in Figure A.3, can be computed as follows. The area occupied by particles, $A_p$, is equal to the number of particles times the area of a particle inside the square unit, and is given by

$$A_p = (N_b n) \times \frac{{R_p}^2}{2} - \frac{{R_p}^2}{2} \approx \frac{2L^2}{n}, \tag{A.2}$$




(a) square structure

(b) a unit of square structure

Figure A.1: A modified two-dimensional bond percolation model

where $N_b$ is the number of bonds in the unit (for square unit, $N_b = 4$) and the length of particle $R_p$ is equal to $L/n$. Since the total area, $A_t$, of the square unit is $L^2$, the filling factor, $\phi$, is

$$\phi = \frac{2}{n}. \tag{A.3}$$

As shown in Equation (A.2), the filling factor is inversely proportional to the number of particles, $n$, in a bond. The value of $n$ can be estimated through experimental or simulation results. The value of $p$ is obtained in our simulation results. Hence, the relation between the volume fraction of filler and conductivity is established.

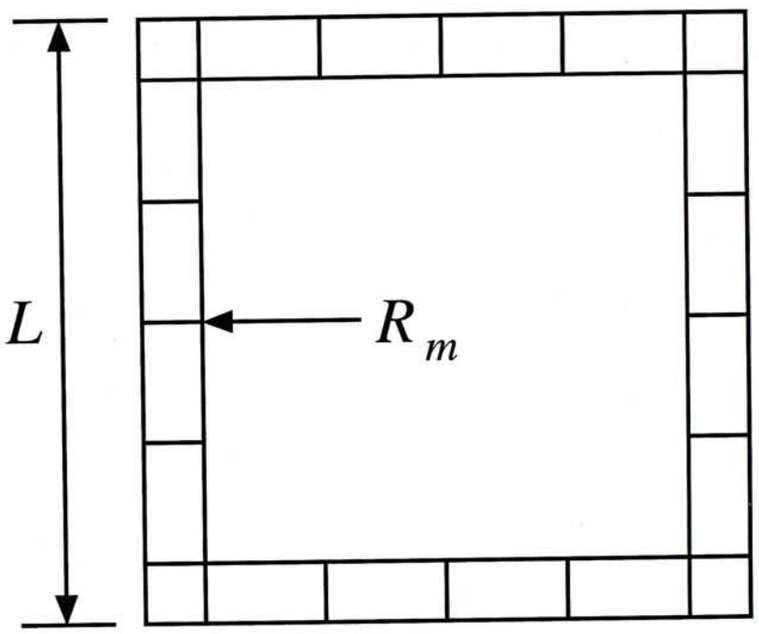



Figure A.2: A unit of square structure.

# Appendix B

# Theory of Elasticity

Stress is a measure of internal forces within a body. This together with strain are the key variables for the determination of stiffness and strength for the determination of stiffness and strength of a material. Consider an isotropic square element under an applied stress in $x$-direction as shown in Figure B.1.

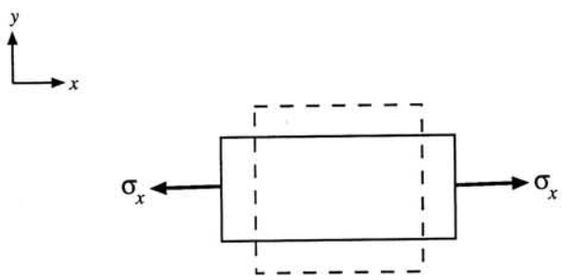



Figure B.1: The deformation of a square element.

We can establish the following stress-strain relations:

$$\varepsilon_x = \frac{1}{E}\sigma_x, \tag{B.1}$$

$$\varepsilon_y = -\frac{\nu}{E}\sigma_x = -\nu\varepsilon_x, \tag{B.2}$$

where $E$ is Young's modulus; $\nu$ is Poisson's ratio is $-\frac{\varepsilon_y}{\varepsilon_x}$. Similarly, the isotropic square element under an applied uniaxial stress $y$-direction only. We get

$$\varepsilon_y = \frac{1}{E}\sigma_y, \tag{B.3}$$

$$\varepsilon_x = -\frac{\nu}{E}\sigma_y = -\nu\varepsilon_y. \tag{B.4}$$

By applying the principle of superposition, we can sum up the contribution of each stress component in Equation (B.1), (B.2), (B.3), (B.4), to resulting strain components. The final stress-strain relation is:

$$\varepsilon_x = \frac{1}{E}\sigma_x - \frac{\nu}{E}\sigma_y, \tag{B.5}$$

$$\varepsilon_y = \frac{1}{E}\sigma_y - \frac{\nu}{E}\sigma_x. \tag{B.6}$$

From Equation (B.5), (B.6), we have

$$E = \frac{{\sigma_x}^2 - {\sigma_y}^2}{\sigma_x\varepsilon_x - \sigma_y\varepsilon_y}, \quad \text{and} \quad \nu = \frac{\sigma_x}{\sigma_y} - E\frac{\varepsilon_x}{\sigma_y}. \tag{B.7}$$

# Appendix C

# Random Number Generator

Simple Linear Congruential Generators (LCGs) obtain the new random number solely from the previous number in the sequence. One might expect that correlation would be reduced by combining more than one previous value. After the simulation, we find that the performance of combining two LCGs is better than than that of simple LCGs. Hence, we do not simply apply on the simulations with a simple LCG function like drand48().

Empirically it has been shown that all the drawbacks of LCGs can be overcome by combining two LCGs.

$$x_i = (A_1 \times x_{i-1} + C_1) \bmod M_1 \tag{C.1}$$

$$y_i = (A_2 \times y_{i-1} + C_2) \bmod M_2 \tag{C.2}$$

$$S_i = (x_i + y_i) \bmod \max(M_1, M_2) \tag{C.3}$$

The good choice of the parameters is

$$A_1 = 40014, C_1 = 0, M_1 = 2147483563 \tag{C.4}$$

$$A_2 = 40692, C_2 = 0, M_2 = 2147483399 \tag{C.5}$$

# Bibliography


[1] D. Stauffer and A. Aharony, *Introduction to Percolation Theory*, London: Taylor & Francis, 1992.

[2] S. H. Kwan, F. G. Shin, and W. L. Tsui, "Dielectric Constant of Silver-Thermosetting Polyester Composites," *J. Mater. Sci. Lett.*, vol. 19, pp. 4093–4098, 1984.

[3] S. R. Broadbent and J. M. Hammersley, "Percolation Processes I. Crystals and Mazes," *Proc. Cambr. Phil. Soc.*, vol. 53, pp. 629–641, 1957.

[4] W. J. Kim, M. Taya, K. Yamada, and N. Kamiya, "Percolation Study on Electrical Resistivity of $SiClSi_3N_4$ Composites with Segregated Distribution," *J. Appl. Phys.*, vol. 83, no. 5, pp. 2593–2598, Mar. 1998.

[5] J. W. Essam, "Percolation Theory," *Rep. Prog. Phys.*, vol. 43, no. 7, pp. 833–912, Jul. 1980.

[6] C. Domb, E. Stoll, and T. Schneider, "Percolation Clusters," *Contemp. Phys.*, vol. 21, no. 6, pp. 577–592, 1980.

[7] L. D. Arcangelis, S. Redner, and A. Coniglio, "Multiscaling Approach in Random Resistor and Random Superconducting Networks," *Phys. Rev. B Solid State*, vol. 34, no. 7, pp. 4656–4672, Oct. 1986.

[8] J. P. Straley, "Critical Exponents for the Conductivity of Random Lattices," *Phys. Rev. B Solid State*, vol. 15, no. 12, pp. 5733–5737, Jun. 1977.

[9] J. Hoshen and R. Kopelman, "Percolation and Cluster Distribution I. Cluster Multiple Labeling Technique and Critical Concentration Algorithm," *Phys. Rev. B Solid State*, vol. 14, no. 8, pp. 3438–3445, Oct. 1976.

[10] I. Balberg, "Recent Developments in Continuum Percolation," *Phil. Mag. B*, vol. 56, no. 6, pp. 991–1003, 1987.

[11] A. Albrecht, S. K. Cheung, K. S. Leung, and C. K. Wong, "Computing Elastic Moduli of Two-Dimensional Random Networks of Rigid and Nonrigid Bonds by Simulated Annealing," *Math. and Comp. in Simulation*, vol. 44, no. 2, pp. 187–215, Sep. 1997.

[12] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[13] V. Cerny, "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulated Algorithm," *J. Optimiz. Theo. Appli.*, vol. 45, no. 1, pp. 41–51, Jan. 1985.

[14] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*, Dordrecht: D. Reidel, 1988.

[15] R. Azencott, *Simulated Annealing : Parallelization Techniques*, New York: Wiley, 1992.

[16] N. Braithwaite and G. Weaver, *Electronic Materials*, Milton Keynes, England: Butterworth, 1990.

[17] C. R. M. Grovernor, *Microelectronic Materials*, Bristol: IOP Press, 1989.

[18] J. C. Maxwell, *A Treatise on Electricity and Magnetism Vol. 1*, New York: Dover Publications, 1954.

[19] J. W. S. Rayleigh, "On the Influence of Obstacles Arranged in Rectangular Order Upon the Properties of a Medium," *Philos. Mag.*, vol. 34, pp. 481–502, 1892.

[20] E. Duering and D. J. Bergman, "Current Distribution on a Three-Dimensional, Bond-Diluted, Random-Resistor Network at the Percolation Threshold," *J. Statist. Phys.*, vol. 60, pp. 363, 1990.

[21] B. G. Streetman, *Solid State Electronic Devices*, Englewood Cliffs, N.J.: Prentice-Hall, 1990.

[22] A. Sinclair and M. Jerrum, "Approximate Counting, Uniform Generation, and Rapidly Mixing Markov Chains," *Inform. Comput.*, vol. 82, no. 1, pp. 93–133, Jul. 1989.

[23] M. Jerrum and A. Sinclair, "Polynomial-Time Approximation Algorithms for the Ising Model," *SIAM J. Comput.*, vol. 22, no. 5, pp. 1087–1116, Oct. 1993.

[24] R. Fogelholm, "The Conductivity of Large Percolation Network Samples," *J. Phys. C: Solid St. Phys.*, vol. 13, no. 23, pp. L571–L574, Aug. 1980.

[25] D. K. Hale, "The Physical Properties of Composite," *J. Mater. Sci.*, vol. 11, no. 11, pp. 2105–2141, Nov. 1978.

[26] W. M. Au and F. G. Shin, "Calculations of Expansion Coefficient of Composite Materials by Numerical Method Based on Effective Medium Theory," *Acta Materiae Compositae Sinica*, vol. 12, no. 2, pp. L571–L574, Jun. 1995.

[27] L. D. Lau and E. M. Lifshitz, *Theory of Elasticity*, New York: Pergamon Press, 1986.

[28] D. J. Bergmann, "Elastic Moduli near Percolation: Universal Ratio and Critical Exponent," *Phys. Rev. B*, vol. 31, no. 3, pp. 1696–1698, Feb. 1985.

[29] D. J. Bergmann, "Elastic Moduli near Percolation in a Two-Dimensional Random Network of Rigid and nonrigid Bonds," *Phys. Rev. B*, vol. 33, no. 3, pp. 2013–2015, Feb. 1986.

[30] D. J. Bergmann, "Universal Poisson's Ratio in a Two-Dimensional Random Network of Rigid and nonrigid Bonds," *Phys. Rev. B*, vol. 34, no. 11, pp. 8199–8201, Dec. 1986.

[31] S. Arbabi and M. Sahimi, "Mechanics of Disordered Solids I. Percolation on Elastic Networks with Central Forces," *Phys. Rev. B*, vol. 47, no. 2, pp. 695–702, Jan. 1993.

[32] M. Sahimi and S. Arbabi, "Mechanics of Disordered Solids II. Percolation on Elastic Networks with Bond-Bending Forces," *Phys. Rev. B*, vol. 47, no. 2, pp. 703–712, Jan. 1993.

[33] S. W. Tsai and H. T. Hahn, *Introduction to Composite Materials*, Westport, Conn.: Technomic Pub., 1980.

[34] G. C. Fox, R. D. Williams, and P. C. Messina, *Parallel Computing Works!*, San Franciso, Calif.: Morgan Kaufmann, 1994.

[35] B. Lewis and D. J. Berg, *Multithreaded Programming with Pthreads*, Upper Saddle River, N.J.: Sun Microsystems Press, 1998.

[36] R. E. Randelman and G. S. Grest, "*N*-City Traveling Salesman Problem - Optimization by Simulated Annealing," *J. Stat. Phys.*, vol. 45, pp. 885–890, 1986.

[37] K. A. Robbins and S. Robbins, *Practical UNIX Programming : A Guide to Concurrency, Communication, and Multithreading*, Upper Saddle River, N.J.: Prentice Hall PTR, 1996.

[38] D. J. Bergman and Y. Kantor, "Critical Properties of an Elastic Fractal," *Phys. Rev. Lett.*, vol. 53, no. 6, pp. 511–514, Aug. 1984.

[39] Y. Kantor and I. Webman, "Elastic Properties of Random Percolating Systems," *Phy. Rev. lett.*, vol. 52, no. 21, pp. 1891–1894, May 1984.

[40] S. Feng and P. N. Sen, "Percolation on Elastic Networks: New Exponent and Threshold," *Phy. Rev. lett.*, vol. 52, no. 3, pp. 216–219, Jan. 1984.

CUHK Libraries
003723626